

137) the reproducible number of the reproducible number field for limiting the number of reproduction of copied programs of the CPTC information, the maximum reproducible time of the maximum reproducible time field for limiting time to reproduce the copied program, and the number and time of reproduction of tape, to thereby process reproduction-impossible.

The copying number limiting step comprises the steps of: comparing (130) the permissible generation of the permissible generational field and the present generation of the present generational field and deciding whether the permissible generation is below the present generation; if the permissible generation is below the present generation, generating (131) an output disable signal to make copying impossible and destroying the control word; and if the permissible generation is not below the present generation, increasing (132) the present invention by '1' and recording the result on cassette tape, and if the permissible generation is not below the present generation, updating the CPTC information in step 133.

The reproduction limiting step comprises the steps of: comparing the reproducible number of the reproducible number field and the reproduction number of tape and deciding (134) whether the reproducible number is below the reproduction number of tape; if the reproducible number is not below the reproduction number of tape, comparing the maximum reproducible time and reproduction time of tape, and deciding (135) whether the maximum reproducible time is below the reproduction time of tape; if the maximum reproducible time is not below reproduction time of tape, turning off (136) an enable erase signal to thereby enable the copied program to be reproduced; if the reproducible number is below the reproduction number of tape or the maximum reproducible time is below the reproduction time of tape, turning on (137) the enable erase signal to make the reproduction of the copied program impossible so that part of or the whole program recorded on cassette tape is erased.

The illegal view/copy protection method for digital broadcasting system embodying the present invention, after the audio/video signal transmission step and audio/video reception step, further comprises a reproduction and rerecording step of: decrypting the bit stream recorded and reproduced on cassette tape, analyzing the CPTC information, deciding whether to allow rerecording, recording the result on cassette tape, filtering the control word, and performing descrambling and decoding to output an audio/video signal.

Referring to Fig. 12, the audio/video reproduction and rerecording step comprises the steps of: filtering (120) the bit stream recorded and reproduced on video tape, and decrypting (121) the CPTC information; analyzing (122 and 123) the CPTC information to generate control words and a signal for controlling the protection of copyright and update the CPTC information; deciding (124) whether to allow recording according to the signal

of controlling the protection of copyright, and recording the scrambled and transmitted bit stream on cassette tape; descrambling and decoding (125 and 126) the transmitted bit stream in control words to output an audio/video signal; and deciding whether to allow post-reproduction according to the signal for controlling the protection of copyright to thereby erase part of or the whole data recorded on cassette tape.

Here, EMM may contain information required for decoding information in order to perform the illegal view/copy protection method of a broadcasting system. In this case, a step of storing and processing the EMM is added in the audio/video reproduction and rerecording step.

In the EMM storing and processing step, in case that the EMM is updated by a broadcasting station for the purpose of copyright protection, the EMM having information required to decode the CPTC information is stored in order to continuously reproduce programs of copied cassette tape.

Here, an ID number indicative of updating the EMM is recorded on cassette tape. The EMM is stored to which the updating state and the ID number of cassette tape are mapped.

The EMM storing and processing step comprises the steps of: storing all EMM to be updated and corresponding ID information; selecting the latest EMM in recording cassette tape; recording a corresponding ID number; and selecting an EMM corresponding to the ID number recorded on cassette tape in reproducing the cassette tape.

As shown in Fig. 13, all EMMs (EMM1, EMM2, EMM3,...) to be updated on the EMM lookup table and corresponding ID information (ID1, ID2, ID3,...) are mapped and stored.

Referring to Figs. 14 and 15, in recording a program on cassette tape, that is, when recording is indicated in the recording/reproduction mode, an ID number corresponding to the latest, the final, EMM, is recorded. Thereafter, in reproducing the cassette tape, that is, when reproduction is indicated in the recording/reproduction mode, an EMM corresponding to the ID number recorded on cassette tape is selected from the EMM lookup table so that the recorded program is reproduced according to the reproducible number of the reproducible number field and the reproduction number recorded on the video tape.

Referring to Fig. 16, an illegal view/copy protection apparatus of digital broadcasting system embodying the present invention comprises a program producing portion 200, distribution medium portion 201, and program receiving portion 202.

Program producing portion 200 offers programs, in which information encrypted both with the control word for scrambling and the CPTC information for prohibiting illegal view/copy, and the audio/video bit stream scrambled in control words are multiplexed to make a program.

Distribution medium portion 201 distributes pro-

grams made in program producing portion 200 through a transmission medium.

Program receiving portion 202 detects and analyzes the CPTC information from the bit stream transmitted from distribution medium portion 201 and the bit stream reproduced from cassette tape, and descrambles and decodes the bit stream transmitted from distribution medium portion 201. The descrambled and decoded bit stream is displayed or recorded on cassette tape.

Program producing portion 200 comprises a control word generator 203 for generating a control word for scrambling, a CPTC generator 204 for generating the CPTC information for prohibiting illegal view/copy, a scrambling portion 206 for scrambling the audio/video bit stream using the control word output from control word generator 203, an encrypting portion 205 for encrypting the control word output from control word generator 203 and the CPTC information output from CPTC generator 204, and an adder 207 for multiplexing the signals output from scrambling portion 206 and encrypting portion 205 and transmitting them to distribution medium portion 201.

Distribution medium portion 201 comprises a broadcasting medium 208 for distributing the program made by program producing portion 200 through cable, satellite or terrestrial broadcasting, and a recording medium 209 for distributing the program made by program producing portion 200 through cassette tape.

Program receiving portion 202 comprises a decrypting portion 210 for decrypting the bit stream transmitted from broadcasting medium 208, a CPTC detecting/analyzing portion 211 for detecting and analyzing the CPTC information from the bit stream output from decrypting portion 210 and recording medium 209, and outputting signals for controlling the control word and illegal view/copy, a descrambling portion 212 for descrambling the bit stream transmitted from broadcasting medium 208 and recording medium 209 and the bit stream reproduced from cassette tape, a decoding portion 213 for decoding and displaying the signal output from descrambling portion 212, and a recording/reproducing portion 214 for recording the bit stream transmitted from broadcasting medium 208 and recording medium 209 according to the signal output from CPTC detecting/analyzing portion 211, and reproducing cassette tape, to thereby output the result to descrambling portion 212 and CPTC detecting/analyzing portion 211.

The operation of an illegal view/copy protection apparatus for a digital broadcasting system embodying the present invention will be described below.

Control word generator 203 generates a control word for scrambling, and CPTC generator 204 generates the CPTC information for prohibiting illegal view/copy. Scrambling portion 206 scrambles the audio/video bit stream using the generated control word. Encrypting portion 205 encrypts the CPTC information output from CPTC generator 204 using the generated control word. The audio/video bit stream scrambled in scrambling por-

tion 206 is multiplexed with the encrypted CPTC information in adder 207. The multiplexed result is transmitted to a reception port through distribution medium portion 201.

The signal output from adder 207 is transmitted to program receiving portion 202 through broadcasting medium 208 such as cable, satellite, and terrestrial broadcastings, or through recording medium 209 made of cassette tape such as rental tape.

The bit stream transmitted through broadcasting medium 208 is decrypted in decrypting portion 210. The CPTC information is detected and analyzed in CPTC detecting/analyzing portion 211 so that signals for controlling the control word and illegal view/copy are output. Here, the bit stream transmitted to cassette tape through recording medium 209 is reproduced in recording/reproducing portion 214 and input to descrambling portion 212 and CPTC detecting/analyzing portion 211. The bit stream transmitted from broadcasting medium 208 and the bit stream reproduced from recording medium 209 through recording/reproducing portion 214 are descrambled in descrambling portion 212 according to the control word output from CPTC detecting/analyzing portion 211. The signal output from descrambling portion 212 is decoded in decoding portion 213 and displayed. The bit stream transmitted from broadcasting medium 208 and recording medium 209 is recorded on cassette tape in a recording/reproducing portion 214 according to the signal output from CPTC detecting/analyzing portion 211.

Data received from program receiving portion 202 and recorded on cassette tape is made up of the scrambled audio/video bit stream and CPTC information. The configuration of the program receiving portion having decrypting portion 210, CPTC detecting/analyzing portion 211, descrambling portion 212, decoding portion 213 and recording/reproducing portion 214 will be explained with reference to Figs. 17, 18, 19, and 20.

One embodiment of the program receiving portion of Fig. 17 receives and processes data transmitted via a broadcasting medium. Specifically, this embodiment performs conditional access and copy protection.

Referring to Fig. 17, the first embodiment of the program receiving portion comprises an IRD 222 for receiving, decoding and descrambling the bit stream transmitted from broadcasting medium 208, outputting analog audio/video data to be displayed and outputting scrambled digital audio/video data to be recorded on cassette tape, a smart card 221 for decrypting the bit stream output from IRD 222, detecting/analyzing the CPTC information, and outputting the control word and signals for controlling illegal view/copy to IRD 222 in order to perform conditional access and copy protection, a DVCR 223 for recording the digital audio/video data and CPTC information scrambled and output from IRD 222 on cassette tape, and reproducing the scrambled digital audio/video data and CPTC information recorded on cassette tape to be output to IRD 222, and a lookup table 224 for,

in case that the EMM is updated by a broadcasting station for the purpose of copyright protection, storing EMM having information required to decode the CPTC information, and outputting CPTC information corresponding in reproduction to smart card 221 in order to continuously reproduce the program of copied cassette tape. Here, lookup table 221 is mapped and processed as shown in Figs. 13, 14 and 15.

The operation of the first embodiment of the program receiving portion will be described below.

In case that a bit stream, that is, a program, is received through a broadcasting medium, the received audio/video data is scrambled digital audio/video data.

The received bit stream is decoded in IRD 222 and decrypted in smart card 221. Its CPTC information is detected and analyzed so that a signal for controlling the control word and illegal view/copy is output to IRD 222.

IRD 222 descrambles the decoded bit stream using the bit stream output from smart card 221 and signals for controlling illegal view/copy. The descrambled bit stream is output to display analog audio/video data. IRD 222 outputs the scrambled digital audio/video data and CPTC information to DVCR 223 in order to record them on cassette tape.

The scrambled digital audio/video data and CPTC information output from IRD 222 is recorded on cassette tape in DVCR 223. They are in turn reproduced in DVCR 223 and processed in the same manner that the bit stream transmitted via the broadcasting medium is descrambled and processed in IRD 222 and smart card 221. The processed result is output to be displayed on a monitor, or output to the DVCR and recopied.

Here, reproduction and recopy are made possible by the data stored in the permissible generational field, present generational field, reproducible number field, and maximum reproducible time field contained in the CPTC information.

Updated EMM is mapped and stored in lookup table 224 so that, when the EMM is updated through a broadcasting signal in a broadcasting station in order to protect copyright, the program of cassette tape copied can be continuously reproduced.

Lookup table 224 reads out the EMM containing information required to decode the CPTC information in reproducing the cassette tape. Corresponding CPTC information is output to smart card 221 to enable reproduction.

Another embodiment of the program receiving portion shown in Fig. 18 is to receive and process data transmitted through a recording medium, for instance, rental tape.

The second embodiment of the program receiving portion, as shown in Fig. 18, comprises a DVCR 232 for detecting/analyzing the CPTC information from the bit stream transmitted from the recording medium, outputting a control word and signals for controlling illegal view/copy, and reproducing scrambled digital audio/video data, and an IRD 231 for receiving the control word

and signals for controlling illegal view/copy output from DVCR 232, descrambling the scrambled digital audio/video data, and outputting analog audio/video data to be displayed or recorded.

5 The second embodiment of the program receiving portion is to perform CPTC detection and processing carried out in the smart card of the first embodiment of the program receiving portion shown in Fig. 17. The operation of the second embodiment of the program receiving portion will be described below.

10 In case that the bit stream is received through the recording medium, the audio/video data reproduced through the DVCR is scrambled digital audio/video data.

15 The bit stream recorded in DVCR 232 is reproduced. Its CPTC information is detected and analyzed so that the control word and signal for controlling illegal view/copy is output to IRD 231. The bit stream reproduced from DVCR 232 is decoded in IRD 231. The decoded bit stream is descrambled according to the control word and signal for controlling illegal view/copy output from DVCR 232 so that analog audio/video data is output to be displayed.

20 IRD 231 outputs the scrambled digital audio/video data and CPTC information to DVCR 232 to record them on cassette tape. The scrambled digital audio/video data and CPTC information output from IRD 231 is recorded on cassette tape and recopied in DVCR 223.

25 Here, reproduction and recopy are made possible by the data stored in the permissible generational field, present generational field, reproducible number field, and maximum reproducible time field contained in the CPTC information.

30 Referring to Fig. 19, still another embodiment of the program receiving portion is to receive and process data transmitted through a recording medium, performing copy protection (CP).

35 As shown in Fig. 19, the third embodiment of the program receiving portion comprises a DVCR 243 for reproducing the scrambled digital audio/video data and CPTC information recorded on cassette tape through a recording medium, and outputting them to IRD 242, an IRD 242 for decoding/descrambling the bit stream transmitted from DVCR 243, and outputting analog audio/video data to be displayed, and a smart card 241 for decrypting the bit stream output from IRD 242, detecting/analyzing the CPTC, and outputting the control word and signals for controlling copying to IRD 222 to thereby perform CP. The operation of the third embodiment of the program receiving portion will be explained below.

40 In case that the bit stream is received via a recording medium, that is, through rental tape, the reproduced audio/video data is scrambled digital audio/video data.

45 The scrambled digital audio/video data and CPTC information reproduced from DVCR 243 are decoded in IRD 242 and decrypted in smart card 241. The CPTC information is detected and analyzed so that the control word and signal for controlling copying are output to IRD 242.

IRD 242 descrambles the decoded bit stream using the CPTC information output from smart card 241 and signals for controlling copying so that analog audio/video data is output to be displayed.

IRD 242 outputs the scrambled digital audio/video data and CPTC information to DVCR 243 in order to record them on cassette tape. The scrambled digital audio/video data and CPTC information output from IRD 242 are recorded on cassette tape in DVCR 243.

Here, reproduction and recopy are made possible by the data stored in the permissible generational field, present generational field, reproducible number field, and maximum reproducible time field contained in the CPTC information.

Referring to Fig. 20, yet another embodiment of the program receiving portion is to receive and process data transmitted through a recording medium, performing conditional access and CP. This embodiment is made in such a manner that in case of using the same CPTC information as the broadcasting medium, the smart card is commonly used.

As shown in Fig. 20, the fourth embodiment of the program receiving portion comprises a DVCR 253 for reproducing the scrambled digital audio/video data and CPTC information recorded on cassette tape through a recording medium, and outputting them to IRD 252, an IRD 252 for decoding/descrambling the bit stream transmitted from DVCR 253, and outputting analog audio/video data to be displayed, and a smart card 251 for decrypting the bit stream output from IRD 252, detecting/analyzing the CPTC, and outputting the control word and signals for controlling copying to IRD 252 to thereby perform CA and CP. The operation of the third embodiment of the program receiving portion will be explained below.

In case that the bit stream is received via a recording medium, that is, through rental tape and the DVCR, the reproduced audio/video data is scrambled digital audio/video data.

The scrambled digital audio/video data and CPTC information reproduced from DVCR 253 are decoded in IRD 252 and decrypted in smart card 251. The CPTC information is detected and analyzed so that the control word and signal for controlling copying are output back to IRD 252.

IRD 252 descrambles the decoded bit stream using the CPTC information output from smart card 251 and signals for controlling illegal view/copy so that analog audio/video data is output to be displayed.

IRD 252 outputs the scrambled digital audio/video data and CPTC information to DVCR 253 in order to record them on cassette tape. The scrambled digital audio/video data and CPTC information output from IRD 222 are recorded on cassette tape in DVCR 253.

Here, reproduction and recopy are made possible by the data stored in the permissible generational field, present generational field, reproducible number field, and maximum reproducible time field contained in the

CPTC information.

IRD 222, 242, or 252 shown in Fig. 17, 19 or 20 is made in the following configuration as shown in Fig. 21.

Referring to Fig. 21, IRD 222, 242 or 252 comprises a recording/digital output controller 262 for decoding the bit stream transmitted from the broadcasting medium and DVCR, outputting to smart card 221, receiving the control word and signals for controlling illegal view/copy output from smart card 221, and controlling the output of the scrambled digital audio/video data for the purpose of recording and displaying; a descrambler 263 for descrambling the scrambled digital audio/video data output from recording/digital output controller 262 according to the control word output from recording/digital output controller 262, and a display processing portion 264 for processing and outputting the digital audio/video data output from descrambler 263 to be displayed. Here, DVCR 265 performs reproduction mainly. DVCR 223 of the program receiving portion of Fig. 18 combines recording therewith. The operation of IRD 266 will be described below.

The signal output to smart card 261 from recording/digital output controller 262 of IRD 266 is ECM, EMM and CPTC information. The signals output from smart card 261 to IRD 266 are the control word used to descramble and display the bit stream, and a signal for controlling copy protection.

Recording/digital output controller 262 communicates with the smart card, performs recording according to the signals of copy protection, outputs them to the digital output port in order to record them in another set, and outputs the control word and bit stream to descrambler 263.

When output to the recording/digital output port, updated ECM, EMM and CPTC information are output in addition to the scrambled data from recording/digital output controller 262 so that a copy different from the original script, that is, the broadcast or rental tape.

The ECM, EMM and CPTC are transmitted in various combinations. For the first combination, the ECM, EMM and CPTC are independently combined. The second combination is that the CPTC is included in the ECM and the EMM is independently combined. The third is that the CPTC is included in the EMM and the ECM is independently combined.

IRD 231 and DVCR 232 of Fig. 18 use the smart card, and additionally requires a CPTC detection and processing portion in the DVCR, which will be shown in Fig. 22.

DVCR 232 comprises a CPTC detecting/processing portion 276 for detecting/analyzing the CPTC information from the bit stream transmitted from recording medium 209, and outputting the control word and signals for illegal view/copy, and a reproducing portion 277 for reproducing the bit stream transmitted from recording medium 209 and outputting it to the IRD.

IRD 231 comprises a digital output controller 272 for receiving the control word and signals for controlling

illegal view/copy output from CPTC detecting/processing portion 276, and controlling the output of the scrambled digital audio/video data output from reproducing portion 277 in order to display them, a descrambler 273 for descrambling the scrambled digital audio/video data output from digital output controller 262 according to the control word output from digital output controller 262, and a display processing portion 274 for processing and outputting the digital audio/video data output from descrambler 273 in order to display them. The operation of IRD 276 and DVCR 275 will be described below.

CPTC detecting/processing portion 276 operates separately when reproducing portion 277 reproduces the scrambled data so that the CPTC information is detected from the cassette tape.

IRD 276 receives the scrambled data, CPTC information and control word from CPTC detecting/processing portion 276 and reproducing portion 277 from DVCR 275. Therefore, for normal descrambling, the scrambled data and control word are supplied to scrambler 273 from digital output controller 272. To the digital output port, only the scrambled data is output. For this reason, in case that the reproduced data is scrambled, copying is made impossible, and vice versa.

Commonly, in order to control tape copying, the depth of generation copy and the reproduction of tape to be copied are used together. As shown in Fig. 7, this yields the effect of controlling the number of copiable tape.

However, in order to allow copying tape to be reproducible as many as a predetermined number or for a predetermined time, it is necessary to perform communication between the smart card and DVCR.

Referring to Fig. 23, tape state information such as the reproduction number of the current tape is transmitted to smart card 261 from DVCR 265. In order to erase the tape, an enable erase signal is transmitted to DVCR 265 from smart card 261, and the erase head of the DVCR operates.

For tape erasing methods, the whole area of tape is erased by the full-width erase head, or only the control track is erased using the control head. In case that the CPTC is contained in the EMM, signals are input and output between the DVCR and smart card.

As the signals input to IRD 266, there are a broadcasting signal transmitted from a broadcasting medium and a signal reproduced from DVCR 265. The broadcasting signal input to IRD 266 is the scrambled digital data and a control signal having the EMM, ECM and CPTC information. The EMM and ECM are required for CA, the CPTC for copyright protection.

The scrambled digital data is input to descrambler 263. The control signal is input to smart card 261 for performing CA and CP. Using the control signal, smart card 261 restores control word CW and outputs it to descrambler 263. Descrambler 263 descrambles it using the control word.

The ECM output from smart card 261 is output to

DVCR 265 or to an external port. This ECM is updated from the ECM input for copyright protection. The output disable signal output from smart card 261 is a signal to instruct IRD 266 to prohibit recording or copying. This signal is input to recording/digital output controller 262. The tape state signal is output to smart card 261 from DVCR 265 in order to inform the state of tape.

The signal output to DVCR 265 from smart card 261 for the purpose of a predetermined-number reproduction or predetermined-time reproduction is an erase enable signal. The signal for allowing recorded and copied tape to be reproducible even though the EMM information of the smart card is changed is an ID signal.

The ID signal is mapped and stored with corresponding EMM in the lookup table of smart card 261. If necessary, the EMM corresponding to the ID signal is output.

As shown in Fig. 24, the smart card comprises an ECM filter 301 for filtering the ECM from the bit stream output from the IRD, a CPTC/tape state signal filter 302 for filtering the CPTC information and the tape state signal indicative of the state of tape from the bit stream output from the IRD, an EMM filter 303 for filtering the EMM from the bit stream output from the IRD, a lookup table 304 for, in case that the EMM is updated for copyright protection by a broadcasting station, storing the previous EMM containing information required to decode the CPTC information, and outputting CPTC information corresponding in reproduction in order to continuously reproduce the program of cassette tape copied, an EMM processing portion 307 for processing the EMM using the EMM output from EMM filter 303 and lookup table 304 and the tape state signal output from CPTC/tape state signal filter 302, a CPTC processing portion 306 for processing the CPTC information using the signals output from CPTC/tape state signal filter 302 and EMM processing portion 307, and a CA processing portion 305 for outputting control word CW using the signals output from ECM filter 301 and EMM processing portion 307.

In case that the CPTC information is contained in the EMM, as shown in Fig. 25, smart card 221 comprises an ECM filter 311 for filtering the ECM from the bit stream output from the IRD, an EMM filter 312 for filtering the EMM containing the EMM from the bit stream output from the IRD, a tape state signal filter 313 for filtering the tape state signal output from the IRD, a lookup table 314 for, in case that the EMM is updated for copyright protection by a broadcasting station, storing the previous EMM containing information required to decode the CPTC information, and outputting CPTC information corresponding in reproduction in order to continuously reproduce the program of cassette tape copied, an EMM processing portion 317 for processing the EMM using the EMM output from EMM filter 312 and lookup table 314 and the tape state signal output from tape state signal filter 313, a CPTC processing portion 316 for processing the CPTC information using the signals

output from EMM filter 312 and tape state signal filter 313, to thereby output ECM, enable erase signal and ID signal, and a CA processing portion 315 for outputting control word CW using the signals output from ECM filter 311 and EMM processing portion 317.

ECM filter 301 or 311, CPTC/tape state signal filter 302, EMM filter 303 or 312, and tape state signal filter 313 extract ECM, CPTC, tape state signal and EMM, respectively. CA processing portion 305 or 315 generates a control word and performs CA. EMM processing portion 307 or 317 outputs the EMM information to CA processing portion 305 or 315 and CPTC processing portion 306 or 316, and additionally stores the received EMM to the lookup table.

In case that the scrambled digital data and encoded CPTC information are recorded on tape and that the EMM information required to decode the CPTC information is changed, the reproduction of tape is made impossible. According to this fact, the previous EMM is stored in a memory such as the EEPROM of the smart card as shown in Figs. 13 and 14, which is the same as described before.

Specifically, the lookup table is divided into two fields and stores ID information and EMM information, as shown in Fig. 13. In recording and copying, the ID information is recorded on tape, as shown in Fig. 14 in order to select corresponding EMM from the ID information recorded in the reproduction of tape.

In other words, referring to Fig. 14, EMM processing portion 307 receives a recording/playback signal indicating that the current DVCR mode is recording or playback, ID, and tape state signal having information of reproduction number of tape, selects a proper EMM from the lookup table, outputs it to CPTC processing portion 306 or 316 and CA processing portion 305 or 315, and transmits the ID information for the purpose of recording and copying to record it on tape.

Referring to Fig. 11, CPTC processing portion 306 or 316 performs copyright protection for recording or copying. The CPTC information or ECM containing the CPTC information is input to output the output disable signal, enable erase signal, and the CPTC or ECM containing the CPTC.

In order to control generation copy, CPTC processing portion 306 or 316, in case that the permissible generation of the permissible generational field is greater than the present generation recorded on tape, the present generational field is increased by 1 and encrypted again. If not, the output disable signal is generated to prohibit recording and copying.

In order to control reproduction, in case that the reproducible number of tape is greater than the reproducible number of the reproducible number field or the maximum reproducible time of the maximum reproducible time field is greater than the current time, CPTC processing portion 306 or 316 generates enable erase signal to operate the erase head of the DVCR.

In case that time delay produced when the CPTC

or the ECM containing the CPTC is encrypted again becomes a problem to solve, CPTC processing portion 306 or 316 transmits the current generation signal to the DVCR and records it on tape, not modifying the CPTC or the ECM containing the CPTC.

The illegal view/copy protection apparatus for a digital broadcasting system embodying the present invention has means for recording and reproducing the reproduction number information of tape in the DVCR in order to implement the predetermined-number reproducibility of recorded or copied tape. Here, the reproduction number information of tape is updated and recorded again during tape reproduction.

As shown in Fig. 26, the DVCR comprises a deck mechanism 406, a recording/reproducing portion 405 for recording digital data on cassette tape according to the deck mechanism and reproducing the digital data recorded on cassette tape, a reproduction number detecting/updating portion 401 for detecting/updating the reproduction number from the digital data reproduced from recording/reproducing portion 405, and outputting it to the IRD in order to rerecord it in recording/reproducing portion 405, a digital data processing portion 402 for processing the digital data reproduced from recording/reproducing portion 405, outputting it to the IRD, and outputting switching position information for recording and reproducing, a recording/playback switching portion 404 for outputting a switching signal for controlling the reproduction number, the reproduction of digital data and the recording of the updated reproduction number using the switching position information output from digital data processing portion 402, and an error correction encoder/decoder 403 for correcting the error of data output from digital data processing portion 402, and encoding and decoding the data to be output to digital data processing portion 402.

In order to update and rerecord the reproduction number information of tape during playback, the reproduction number information of tape is recorded using an encoding algorithm. Otherwise, the information is recorded as clear data not encoded.

The recording position of the reproduction number information of tape uses part of audio, control and video tracks. For error correction to the reproduction number information of tape, a repetition coding is employed. The operation of the DVCR will be described below.

When reproduced by recording/reproducing portion 405 with the cassette tape loaded on deck mechanism 406, the reproduced digital data is input to reproduction number detecting/updating portion 401 and digital data processing portion 402 so that its reproduction number is detected and the digital data is processed and output.

The reproduction number detected in reproduction number detecting/updating portion 401 is updated, that is, increased by 1, and applied to recording/reproducing portion 405.

Digital data processing portion 402 applies the reproduced digital data output from recording/reproducing

portion 405 to error correction encoder/decoder 403 to perform error correction, encoding and decoding. The result is output to the IRD to be displayed or recorded. At the same time, the switching position information is output to recording/reproducing switching portion 404 in order to output a switching signal.

The switching signal output from recording/reproducing switching portion 404 controls recording/reproducing portion, to thereby record the updated reproduction number output from reproduction number detecting/ updating portion 401, that is, the reproduction number added by 1, on tape.

Recording/reproducing switching portion 404 controls the reproduction number, the reproduction of digital data recorded on tape, and the recording of the updated reproduction number.

In another method of implementing the predetermined-number reproducibility of recorded or copied tape, an identifier is given to all tape used for a user to record broadcast programs, and the identifier given to tape and the reproducibility number information of tape corresponding to the identifier are handled together in the smart card.

Here, the smart card has a memory device which can be updated, such as EEPROM. The identifier and corresponding reproducible number information are stored in the memory device. For every reproduction of tape, the reproducible number information is updated and whether to playback is determined.

In conclusion, the described embodiments have the following advantages.

First, by adding CPTC information to data supplied, and by allowing a digital program to be normally viewed only when a CPTC detecting/analyzing means and descrambling/decrypting means are present at the receiving stage, illegal viewing is prohibited.

Second, to enhance copyright protection, data recorded on cassette tape is always scrambled digital data, and its CPTC information is encrypted to be recorded on cassette tape. A code for prohibiting viewable data from being restored from the cassette tape only with the scrambled data and CPTC information, and allowing the data to be viewable is provided in a device excluding the cassette tape. Otherwise, restoring of viewable data is made possible only with the scrambled data and CPTC information, making illegal copy impossible.

Third, using a method of restoring the viewable data only with the scrambled digital data and CPTC, rental tape is made to supply tape. Otherwise, using a method of prohibiting the viewable data from being restored only with the scrambled digital data and CPTC, rental tape is made to supply tape and smart card peculiar to a program provider as one set. Using the smart card for broadcasting medium, the rental tape is made to prohibit the viewable data from being restored only with the scrambled digital data and CPTC. Among the three methods of supplying tape only, one method is selected. Digital hardware for reproducing the data outputs only

the scrambled digital data to an external port, making impossible the restoring of viewable data from the output data, without the smart card.

Fourth, the described embodiment prohibits illegal recording and copying of a program protected by copyright law, collects fee for recording or copying, and freely controls the reproducible number of copied tape which can be made from a program supplied by a program supplier, protecting copyright.

Fifth, the described embodiment can be used as a copyright protection system having a high security and multifunction with respect to a program through a broadcasting medium such as satellite and terrestrial broadcastings, or, at the same time, as a copy protection system having a high security to a program through a recording medium such as rental tape.

Sixth, the described embodiment is employed to digital hardware such as broadcasting receiver and digital VCR, to thereby perfectly protect a program supplier's copyright and activates digital media because of various software supplied through the digital media.

Claims

1. An illegal view/copy protection method for a digital broadcasting system comprising:

an audio/video signal transmission step for multiplexing and transmitting audio/video bit stream scrambled in control words and information where the control words and CPTC information for illegal view/copy protection are encrypted; and

an audio/video reception step for decrypting the transmitted bit stream to analyze the CPTC information and control words, deciding whether recording is allowed or not to be recorded on cassette tape, and using the control words, performing descrambling and decoding to output audio/video signals to a monitor.

2. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 1, wherein said CPTC information is formatted in a generational copy control field for limiting the number of copy available, and a reproducibility control field for limiting the reproduction of a copied program.

3. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 2, wherein said CPTC information is formatted further containing a descrambling information field where part of the control words for descrambling are recorded.

4. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 2, wherein said CPTC information is formatted further contain-

- ing a CA field where CA information for conditional access is recorded.
5. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 2, wherein said generational copy control field is made up of a permissible generational field for limiting the number of copy permissible and a present generational field for indicating the present generation of a program copied.
6. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 2, wherein said reproduction control field is made up of a reproducible number field for limiting the number of reproducing a copied program, and a maximum reproducible time field for limiting time to reproduce the copied program.
7. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 1, wherein the data recorded on cassette tape contains scrambled audio/video bit stream and CPTC information.
8. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 7, wherein said CPTC information is overwritten on the scrambled audio/video bit stream for the error effect and recorded on cassette tape.
9. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 7, wherein said CPTC information is recorded on a portion of any of the audio track of cassette tape, the control track of cassette tape, or the video track of cassette tape.
10. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 1, wherein said audio/video signal transmission step comprises the steps of: encoding the audio/video bit stream; generating a control word for scrambling; scrambling for the encoded audio/video bit stream using the generated control word; generating CPTC information for illegal view/copy protection; encrypting for encrypting the control word and CPTC information; and multiplexing and transmitting the scrambled audio/video bit stream and encrypted CPTC information.
11. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 1, wherein said audio/video signal transmission step comprises the steps of:
- encoding the audio/video bit stream; generating a control word for scrambling; scrambling for the encoded audio/video bit stream using the generated control word; generating CPTC information for illegal view/copy protection; generating conditional access information for conditional reception; encrypting for encrypting the CPTC information and CA information; and multiplexing and transmitting the scrambled audio/video bit stream and encrypted CPTC information and conditional access information.
12. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 1 or claim 11, wherein said audio/video reception step comprises the steps of:
- filtering the transmitted bit stream and decrypting the CPTC information; analyzing the CPTC information to generate a control word and a signal for controlling the protection of copyright and to update the CPTC information; deciding whether to allow recording according to the signal for controlling the protection of copyright to record the scrambled and transmitted bit stream on cassette tape; and descrambling and decoding the transmitted bit stream in the control word and outputting an audio/video signal.
13. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 12, wherein said all of the control word is contained in the CPTC information.
14. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 1, wherein said bit stream transmitted contains ECM and EMM.
15. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 14, wherein said audio/video reception step comprises the steps of:
- filtering the transmitted bit stream and decrypting the CPTC information and control word; filtering the control word; analyzing the CPTC information to generate a control word and a signal for controlling the protection of copyright and to update the CPTC information; deciding whether to allow recording according to the signal for controlling the protection of copyright to record the scrambled and transmit-

ted bit stream on cassette tape; and descrambling and decoding the transmitted bit stream in control words and outputting an audio/video signal.

16. An illegal view/copy protection method for a digital broadcasting system as claimed in any of claims 12, 14 or 15, wherein said CPTC information analyzing step comprises the steps of:

generating a control word;
 detecting a permissible generation of a permissible generational field for limiting the available number of copy of a program of the CPTC information and the present generation of the present generational field indicating the present generation of the program copied, to thereby perform copy-impossible and update the CPTC information; and
 detecting the reproducible number of the reproducible number field for limiting the number of reproduction of copied programs of the CPTC information, the maximum reproducible time of the maximum reproducible time field for limiting time to reproduce the copied program, and the number and time of reproduction of tape, to thereby process reproduction-impossible.

17. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 12 or claim 16, wherein said copying number limiting step comprises the steps of:

comparing the permissible generation of the permissible generational field and the present generation of the present generational field and deciding whether the permissible generation is below the present generation;
 if the permissible generation is below the present generation, generating an output disable signal to make copying impossible and destroying the control word; and
 if the permissible generation is not below the present generation, increasing the present invention by '1' and recording the result on cassette tape.

18. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 17, wherein said copying number limiting step further comprises the step of, if the permissible generation is not below the present generation, updating the CPTC information.

19. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 16 or claim 17, wherein said reproduction limiting step comprises the steps of:

comparing the reproducible number of the reproducible number field and the reproduction number of tape and deciding whether the reproducible number is below the reproduction number of tape;

if the reproducible number is not below the reproduction number of tape, comparing the maximum reproducible time and reproduction time of tape, and deciding whether the maximum reproducible time is below the reproduction time of tape;

if the maximum reproducible time is not below reproduction time of tape, turning off an enable erase signal to thereby enable the copied program to be reproduced; and

if the reproducible number is below the reproduction number of tape or the maximum reproducible time is below the reproduction time of tape, turning on the enable erase signal to make the reproduction of the copied program impossible so that part of or the whole program recorded on cassette tape is erased.

20. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 14 or claim 15, wherein part of the control word is contained in the CPTC information.

21. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 20, wherein the remainder of the control word is contained in the ECM.

22. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 20, wherein the remainder of the control word is contained in the EMM.

23. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 14 or claim 15, wherein the whole control word is contained in the ECM.

24. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 14 or claim 15, wherein the whole control word is contained in the EMM.

25. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 14, further comprising a reproduction and rerecording step of: decrypting the bit stream recorded and reproduced on cassette tape, analyzing the CPTC information, deciding whether to allow rerecording, recording the result on cassette tape, filtering the control word, and performing descrambling and decoding to output an audio/video signal.

26. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 25, wherein said audio/video reproduction and rerecording step comprises the steps of:
- 5 filtering the bit stream recorded and reproduced on video tape, and decrypting the CPTC information;
 - analyzing the CPTC information to generate control words and a signal for controlling the protection of copyright and update the CPTC information;
 - 10 deciding whether to allow recording according to the signal of controlling the protection of copyright, and recording the scrambled and transmitted bit stream on cassette tape; and
 - 15 descrambling and decoding the transmitted bit stream in control words to output an audio/video signal.
27. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 26, wherein said audio/video reproduction and rerecording step comprises the step of deciding whether to allow post-reproduction according to the signal for controlling the protection of copyright to thereby erase part of or the whole data recorded on cassette tape.
28. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 25, wherein said EMR contains information required for decoding information
29. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 28, further comprising the step of storing and processing EMM in which, in case that the EMM is updated by a broadcasting station for the purpose of copyright protection, the EMM having information required to decode the CPTC information is stored in order to continuously reproduce programs of copied cassette tape.
30. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 29, wherein an ID number indicative of updating the EMM is recorded on said cassette tape.
31. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 30, wherein the EMM is stored to which the updating state and the ID number of cassette tape are mapped.
32. An illegal view/copy protection method for a digital broadcasting system as claimed in claim 31, wherein said EMM storing and processing step comprises the steps of:
- storing all EMM to be updated and corresponding ID information;
 - selecting the latest EMM in recording cassette tape;
 - 5 recording a corresponding ID number; and
 - selecting an EMM corresponding to the ID number recorded on cassette tape in reproducing the cassette tape.
33. An illegal view/copy protection apparatus for a digital broadcasting system comprising:
- a program producing portion for multiplexing information encrypted both with the control word for scrambling and the CPTC information for prohibiting illegal view/copy, and the audio/video bit stream scrambled in control words, to thereby make a program;
 - a distribution medium portion for distributing programs made in said program producing portion through a transmission medium; and
 - 20 a program receiving portion for detecting and analyzing the CPTC information from the bit stream transmitted from said distribution medium portion and the bit stream reproduced from cassette tape, and descrambling and decoding the bit stream transmitted from said distribution medium portion.
34. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 33, wherein said program producing portion comprising:
- a control word generator for generating a control word for scrambling;
 - a CPTC generator for generating the CPTC information for prohibiting illegal view/copy;
 - a scrambling portion for scrambling the audio/video bit stream using the control word output from said control word generator;
 - an encrypting portion for encrypting the control word output from said control word generator and the CPTC information output from said CPTC generator; and
 - an adder for multiplexing the signals output from said scrambling portion and encrypting portion and transmitting them to said distribution medium portion.
35. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 33, wherein said distribution medium portion comprises:
- a broadcasting medium for distributing the program made by said program producing portion through cable, satellite or terrestrial broadcast-

ing; and
 a recording medium for distributing the program made by said program producing portion through cassette tape.

36. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 35, wherein said program receiving portion comprises:

a decrypting portion for decrypting the bit stream transmitted from said broadcasting medium;

a CPTC detecting/analyzing portion for detecting and analyzing the CPTC information from the bit stream output from said decrypting portion and recording medium, and outputting signals for controlling the control word and illegal view/copy;

a descrambling portion for descrambling the bit stream transmitted from said broadcasting medium and recording medium and the bit stream reproduced from cassette tape;

a decoding portion for decoding and displaying the signal output from said descrambling portion; and

a recording/reproducing portion for recording the bit stream transmitted from said broadcasting medium and recording medium according to the signal output from said CPTC detecting/analyzing portion, and reproducing cassette tape, to thereby output the result to said descrambling portion and CPTC detecting/analyzing portion.

37. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 33, wherein said CPTC information is formatted in a generational copy control field for limiting the number of copy available, and a reproducibility control field for limiting the reproduction of a copied program.

38. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 37, wherein said CPTC information is formatted further containing a descrambling information field where the whole or part of the control words for descrambling are recorded.

39. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 37, wherein said CPTC information is formatted further containing a CA field where CA information for conditional access is recorded.

40. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 37, wherein said generational copy control field is

made up of a permissible generational field for limiting the number of copy permissible and a present generational field for indicating the present generation of a program copied.

41. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 37, wherein said reproduction control field is made up of a reproducible number field for limiting the number of reproducing a copied program, and a maximum reproducible time field for limiting time to reproduce the copied program.

42. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 33, wherein the data recorded on cassette tape contains scrambled audio/video bit stream and CPTC information.

43. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 42, wherein said CPTC information is overwritten on the scrambled audio/video bit stream for the error effect and recorded on cassette tape.

44. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 42, wherein said CPTC information is recorded on a portion of any of the audio track of cassette tape, the control track of cassette tape, or the video track of cassette tape.

45. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 33, wherein said all of the control word is contained in the CPTC information.

46. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 33, wherein said bit stream transmitted contains ECM and EMM.

47. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 46, wherein part of the control word is contained in the CPTC information.

48. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 47, wherein the remainder of the control word is contained in the ECM.

49. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 47, wherein the remainder of the control word is contained in the EMM.

50. An illegal view/copy protection apparatus for a dig-

- ital broadcasting system as claimed in claim 46, wherein the whole control word is contained in the ECM.
51. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 46, wherein the whole control word is contained in the EMM. 5
52. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 46, wherein said program receiving portion comprises:
- an IRD for receiving, decoding and descrambling the bit stream transmitted from said broadcasting medium, outputting analog audio/video data to be displayed and outputting scrambled digital audio/video data to be recorded on cassette tape; and 15
- a smart card for decrypting the bit stream output from said IRD, detecting/analyzing the CPTC information, and outputting the control word and signals for controlling illegal view/copy to said IRD in order to perform conditional access and copy protection. 20
53. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 52, wherein said program receiving portion further comprises a lookup table for, in case that the EMM is updated by a broadcasting station for the purpose of copyright protection, storing EMM having information required to decode the CPTC information, and outputting CPTC information corresponding in reproduction to said smart card in order to continuously reproduce the program of copied cassette tape. 30
54. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 52, wherein said program receiving portion further comprises a DVCR for recording the digital audio/video data and CPTC information scrambled and output from said IRD on cassette tape, and reproducing the scrambled digital audio/video data and CPTC information recorded on cassette tape to be output to said IRD. 40
55. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 54, wherein said smart card comprises:
- an ECM filter for filtering the ECM from the bit stream output from said IRD; 45
- a CPTC/tape state signal filter for filtering the CPTC information and the tape state signal indicative of the state of tape from the bit stream output from said IRD; 50
- an EMM filter for filtering the EMM from the bit stream output from said IRD; 55
- a lookup table for, in case that the EMM is updated for copyright protection by a broadcasting station, storing the previous EMM containing information required to decode the CPTC information, and outputting CPTC information corresponding in reproduction in order to continuously reproduce the program of cassette tape copied;
- an EMM processing portion for processing the EMM using the EMM output from said EMM filter and lookup table and the tape state signal output from said CPTC/tape state signal filter;
- a CPTC processing portion for processing the CPTC information using the signals output from said CPTC/tape state signal filter and EMM processing portion; and
- a CA processing portion for outputting control word CW using the signals output from said ECM filter and EMM processing portion.
56. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 54, wherein said smart card comprises:
- an ECM filter for filtering the ECM from the bit stream output from said IRD;
- an EMM filter for filtering the EMM containing the EMM from the bit stream output from said IRD;
- a tape state signal filter for filtering the tape state signal output from said IRD;
- a lookup table for, in case that the EMM is updated for copyright protection by a broadcasting station, storing the previous EMM containing information required to decode the CPTC information, and outputting CPTC information corresponding in reproduction in order to continuously reproduce the program of cassette tape copied;
- an EMM processing portion for processing the EMM using the EMM output from said EMM filter and lookup table and the tape state signal output from said tape state signal filter;
- a CPTC processing portion for processing the CPTC information using the signals output from said EMM filter and tape state signal filter, to thereby output ECM, enable erase signal and ID signal; and
- a CA processing portion for outputting control word CW using the signals output from said ECM filter and EMM processing portion.
57. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 54, wherein said DVCR comprises:

a deck mechanism;
 a recording/reproducing portion for recording digital data on cassette tape according to said deck mechanism and reproducing the digital data recorded on cassette tape;
 a reproduction number detecting/updating portion for detecting/updating the reproduction number from the digital data reproduced from said recording/reproducing portion, and outputting it to said IRD in order to rerecord it in said recording/reproducing portion;
 a digital data processing portion for processing the digital data reproduced from said recording/reproducing portion, outputting it to said IRD, and outputting switching position information for recording and reproducing;
 a recording/playback switching portion for outputting a switching signal for controlling the reproduction number, the reproduction of digital data and the recording of the updated reproduction number using the switching position information output from said digital data processing portion; and
 an error correction encoder/decoder for correcting the error of data output from said digital data processing portion, and encoding and decoding the data to be output to said digital data processing portion.

58. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 35, wherein said program receiving portion comprises:

a DVCR for detecting/analyzing the CPTC information from the bit stream transmitted from said recording medium, outputting a control word and signals for controlling illegal view/copy, and reproducing scrambled digital audio/video data; and
 an IRD for receiving the control word and signals for controlling illegal view/copy output from said DVCR 232, descrambling the scrambled digital audio/video data, and outputting analog audio/video data to be displayed or recorded.

59. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 58, wherein said DVCR comprises:

a CPTC detecting/processing portion for detecting/analyzing the CPTC information from the bit stream transmitted from said recording medium, and outputting the control word and signals for illegal view/copy; and
 a reproducing portion for reproducing the bit stream transmitted from said recording medium and outputting it to said IRD.

60. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 59, wherein said IRD comprises:

a digital output controller for receiving the control word and signals for controlling illegal view/copy output from said CPTC detecting/processing portion, and controlling the output of the scrambled digital audio/video data output from said reproducing portion in order to display them;
 a descrambler for descrambling the scrambled digital audio/video data output from said digital output controller according to the control word output from said digital output controller; and
 a display processing portion for processing and outputting the digital audio/video data output from said descrambler in order to display them.

61. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 35, wherein said program receiving portion comprises:

a DVCR for reproducing the scrambled digital audio/video data and CPTC information recorded on cassette tape through a recording medium, and outputting them to said IRD;
 an IRD for decoding/descrambling the bit stream transmitted from said DVCR, and outputting analog audio/video data to be displayed; and
 a smart card for decrypting the bit stream output from said IRD, detecting/analyzing the CPTC, and outputting the control word and signals for controlling copying to said IRD to thereby perform copy protection and/or conditional access.

62. An illegal view/copy protection apparatus for a digital broadcasting system as claimed in claim 54 or claim 61, wherein said IRD comprises:

a recording/digital output controller for decoding the bit stream transmitted from the broadcasting medium and DVCR, outputting to said smart card, receiving the control word and signals for controlling illegal view/copy output from said smart card, and controlling the output of the scrambled digital audio/video data for the purpose of recording and displaying;
 a descrambler for descrambling the scrambled digital audio/video data output from said recording/digital output controller according to the control word output from said recording/digital output controller; and
 a display processing portion for processing and outputting the digital audio/video data output from said descrambler to be displayed.

FIG. 1
(conventional art)

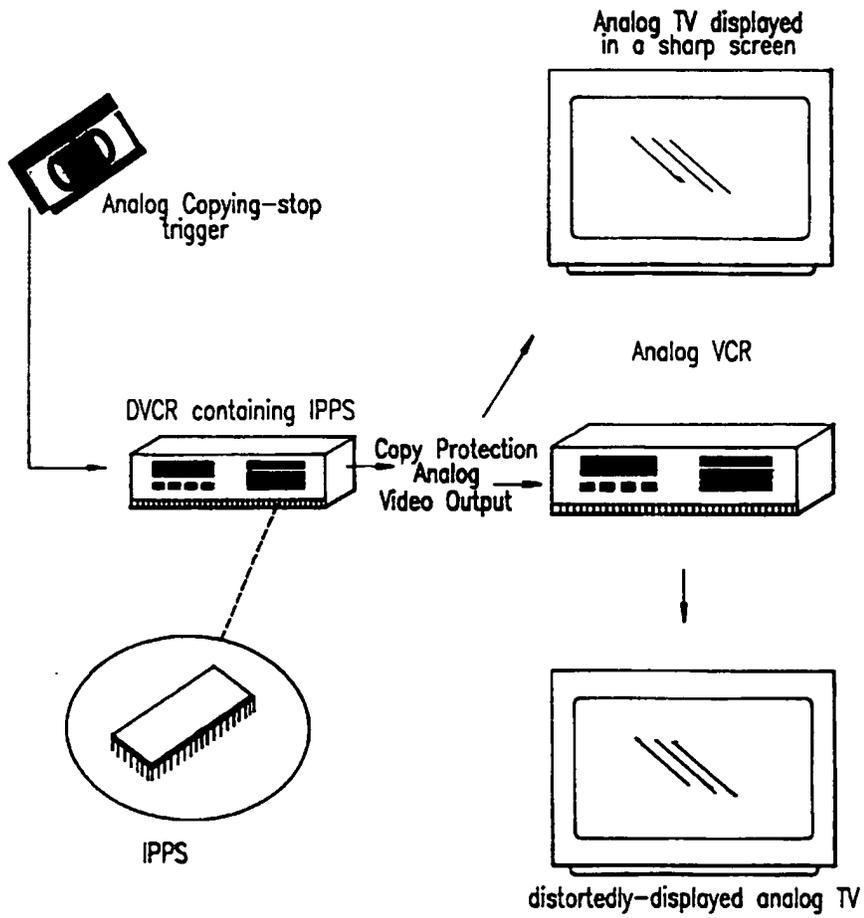
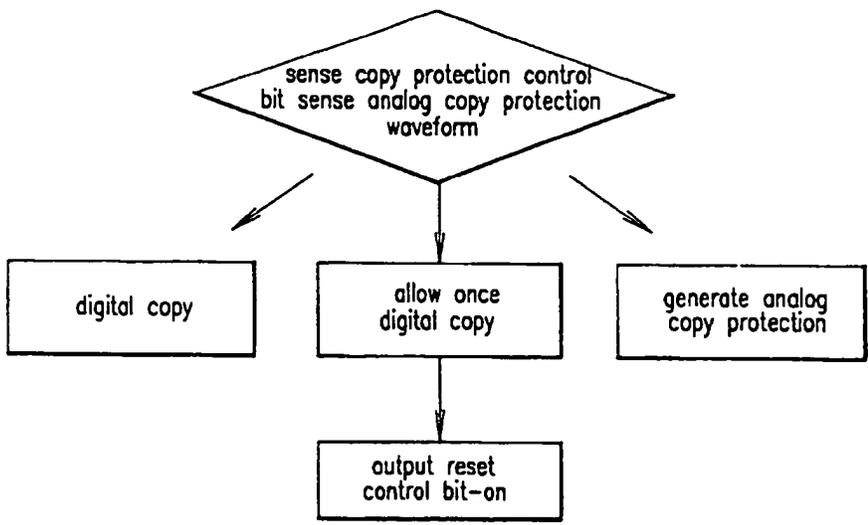


FIG.2
(conventional art)



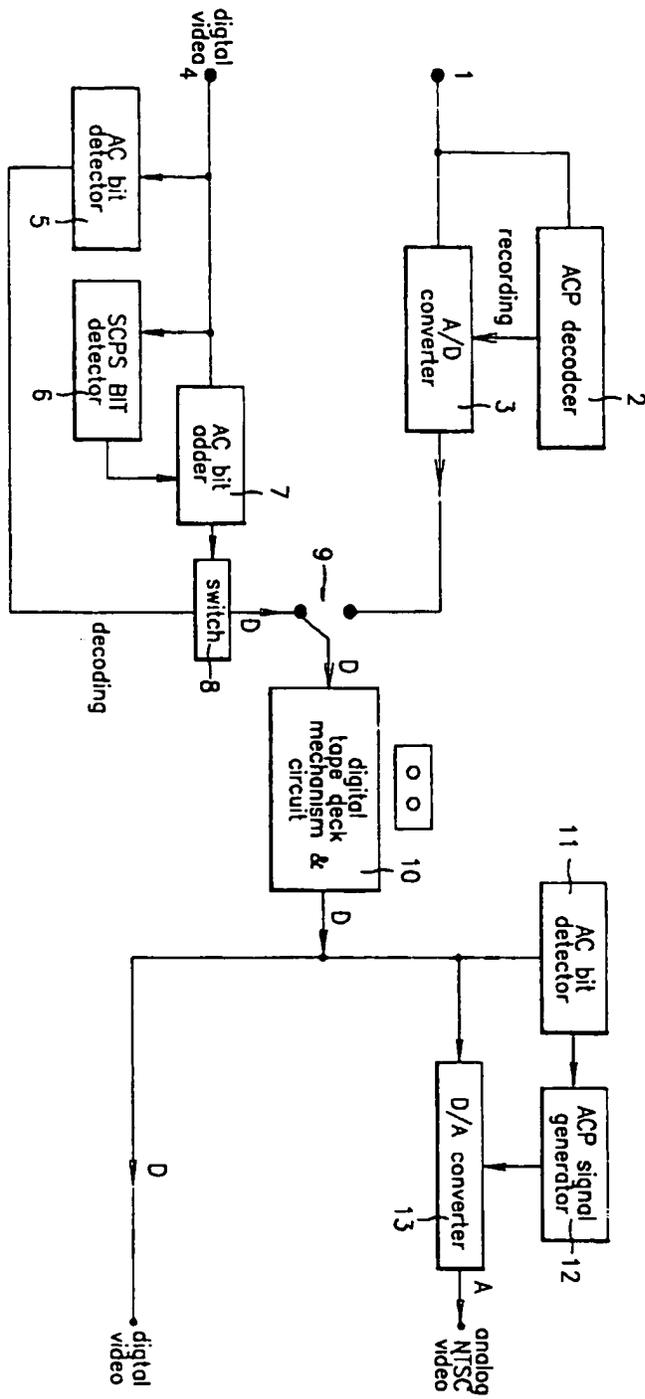


FIG. 3

FIG. 4

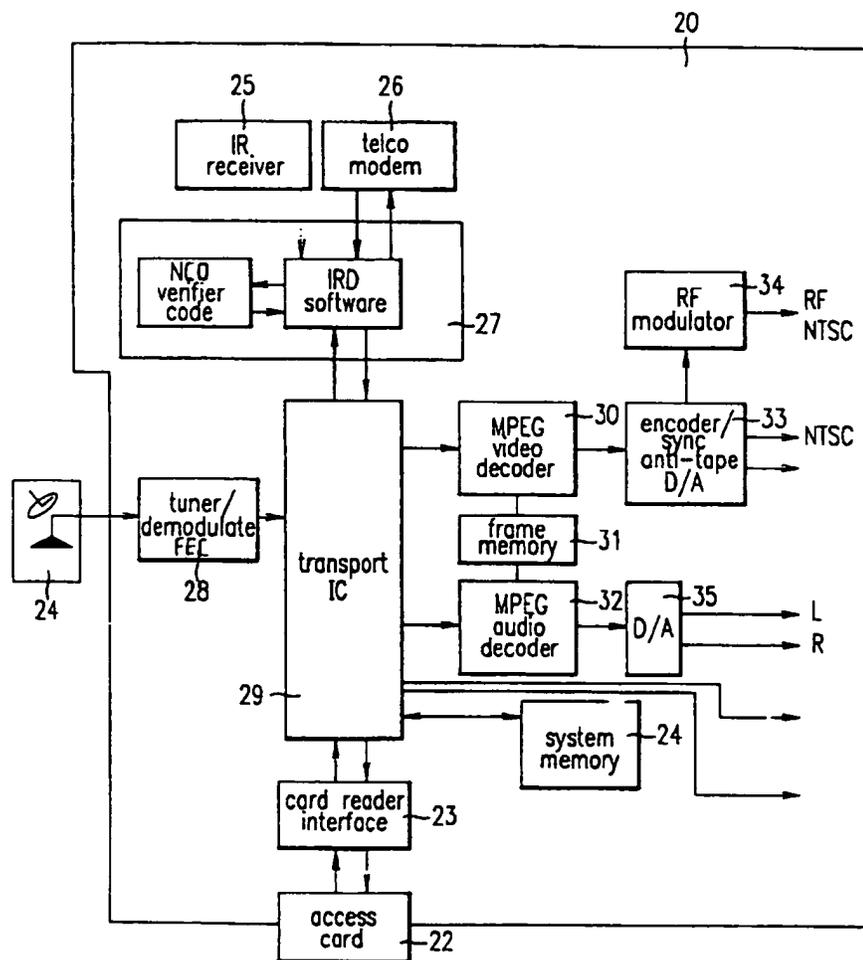
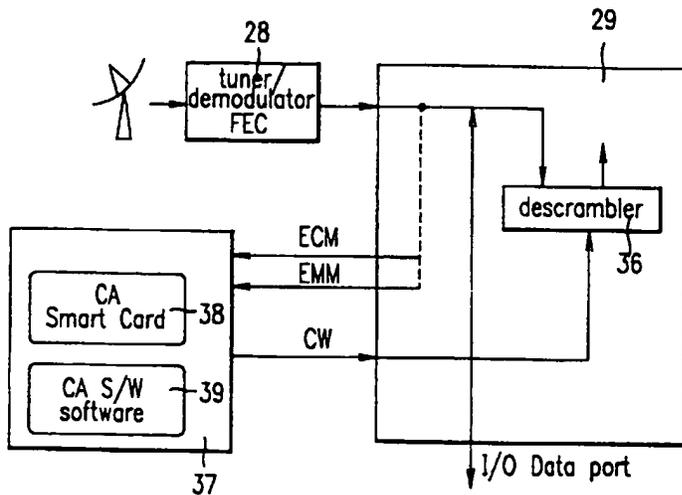
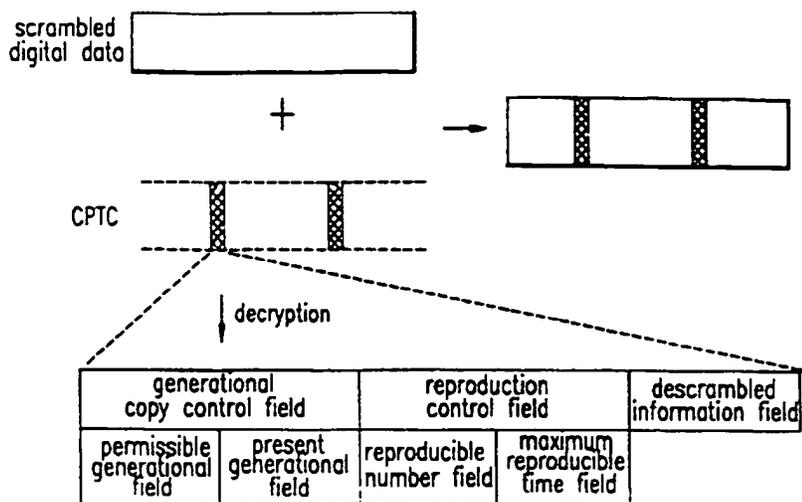


FIG. 5



F I G.6a



F I G.6b

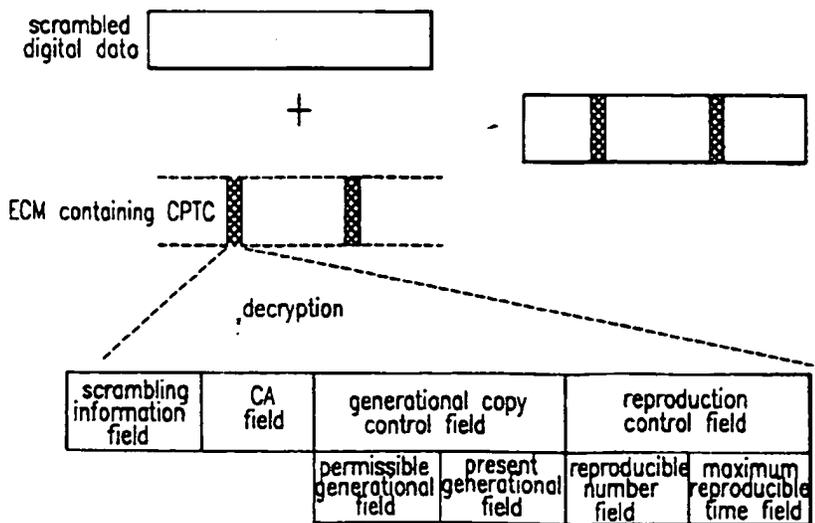


FIG. 7

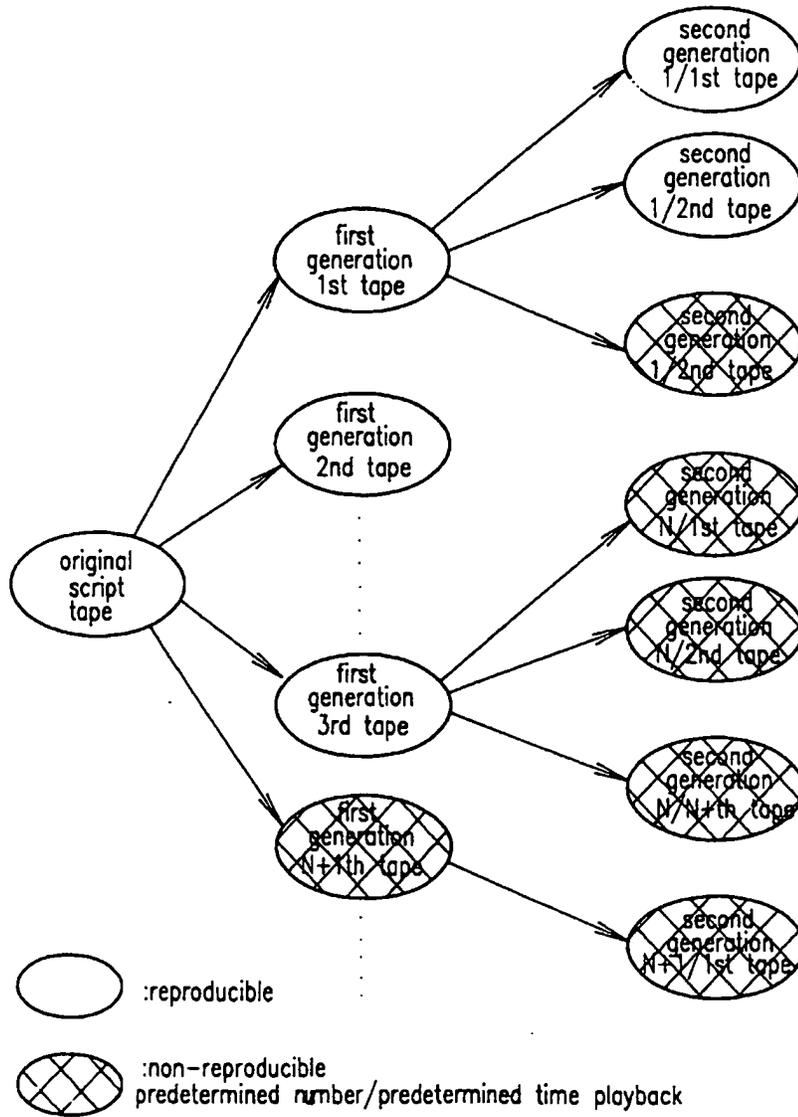


FIG. 8a

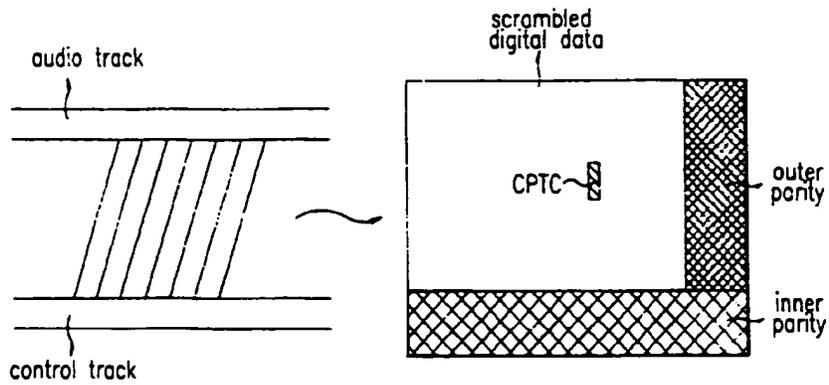


FIG. 8b

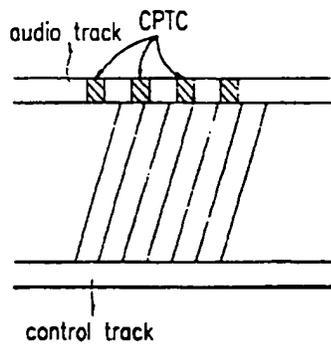


FIG. 8c

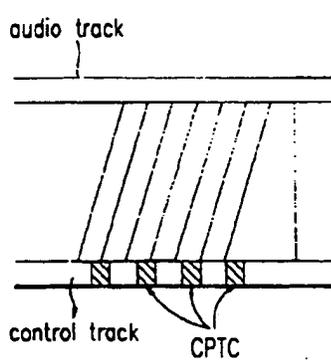


FIG. 8d

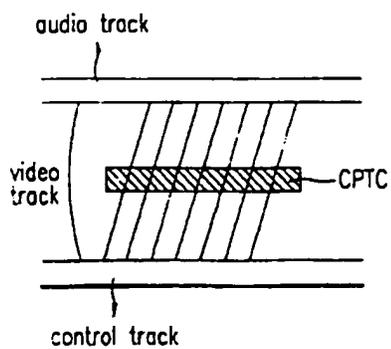


FIG. 9

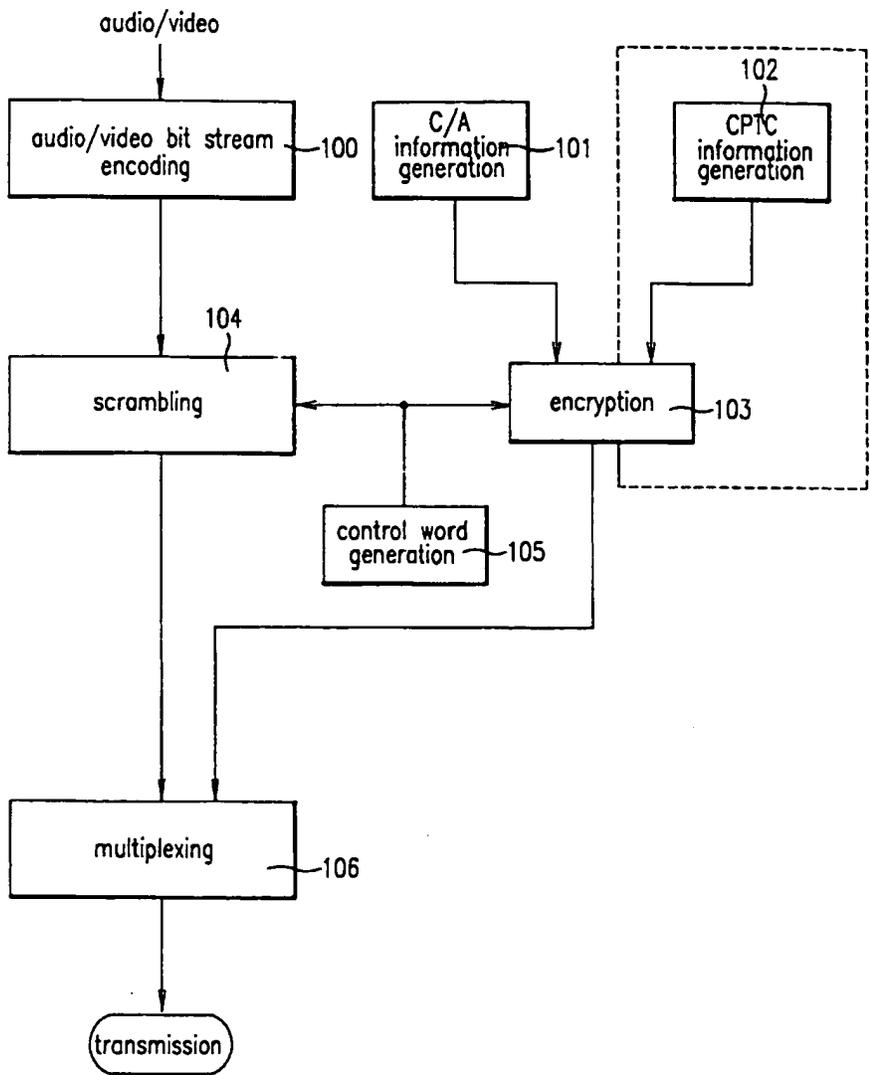


FIG. 10

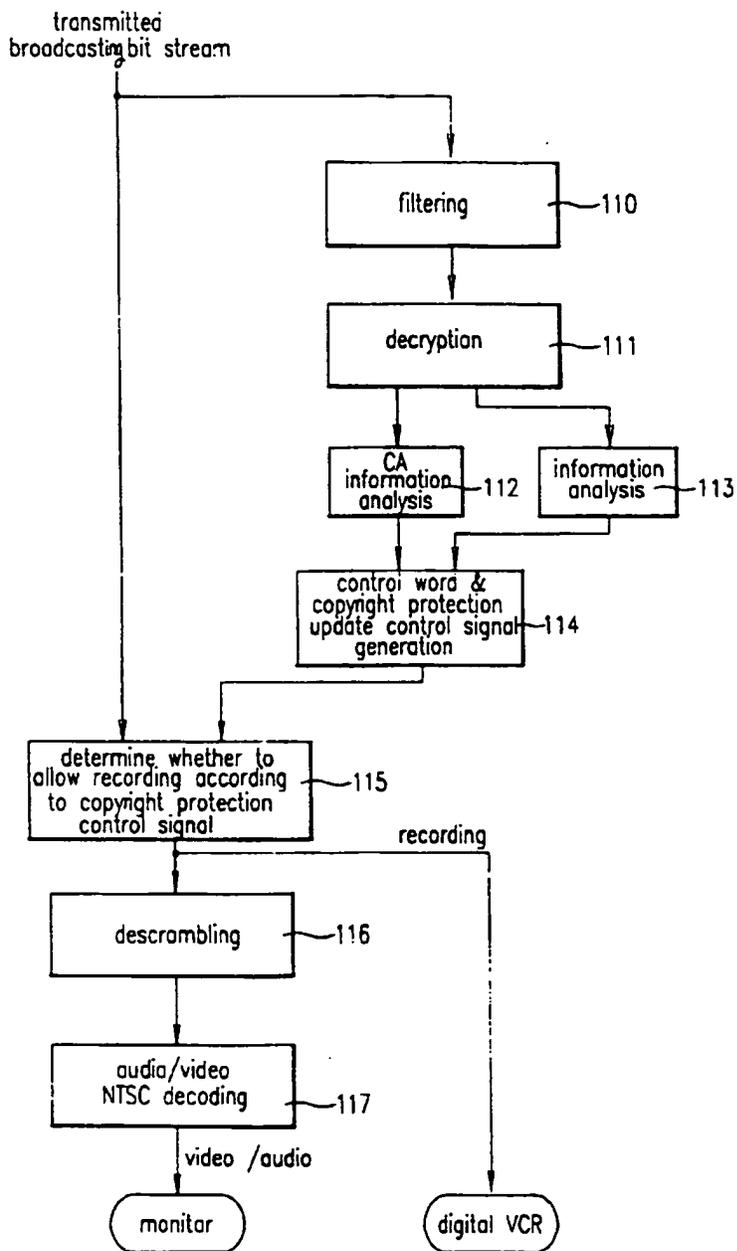


FIG. 11

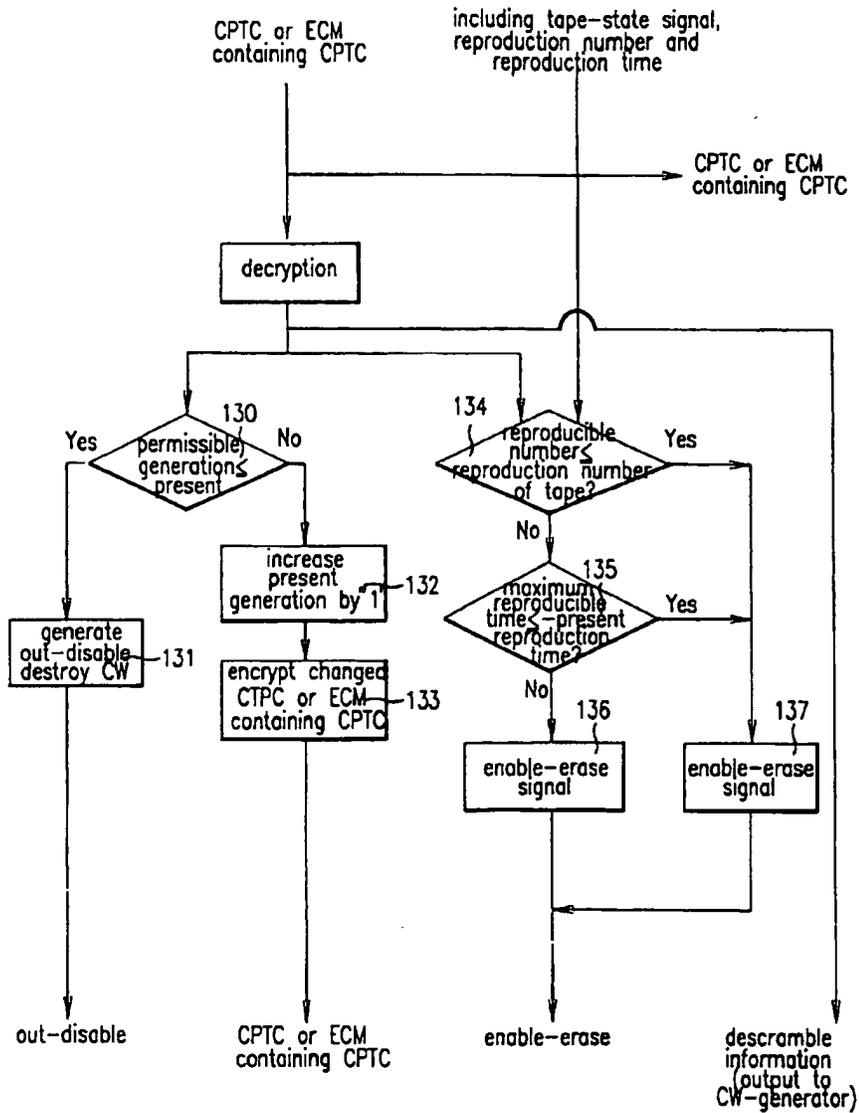


FIG. 12

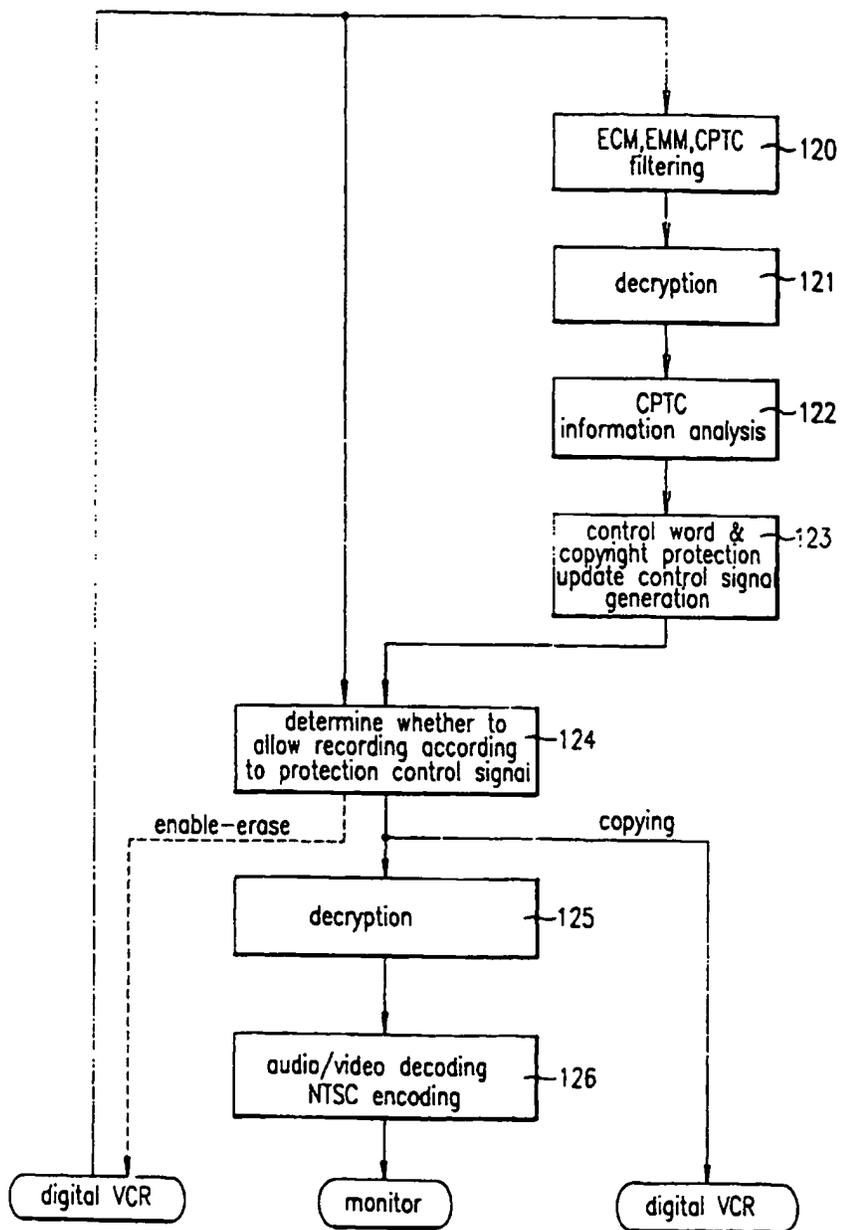


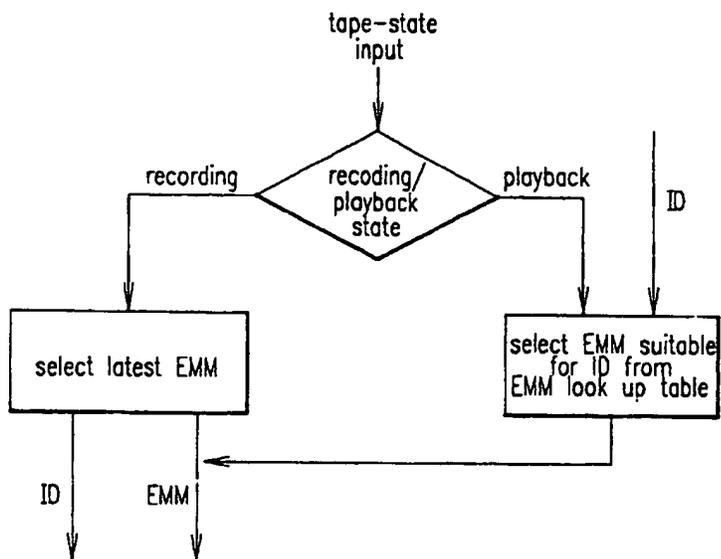
FIG. 13

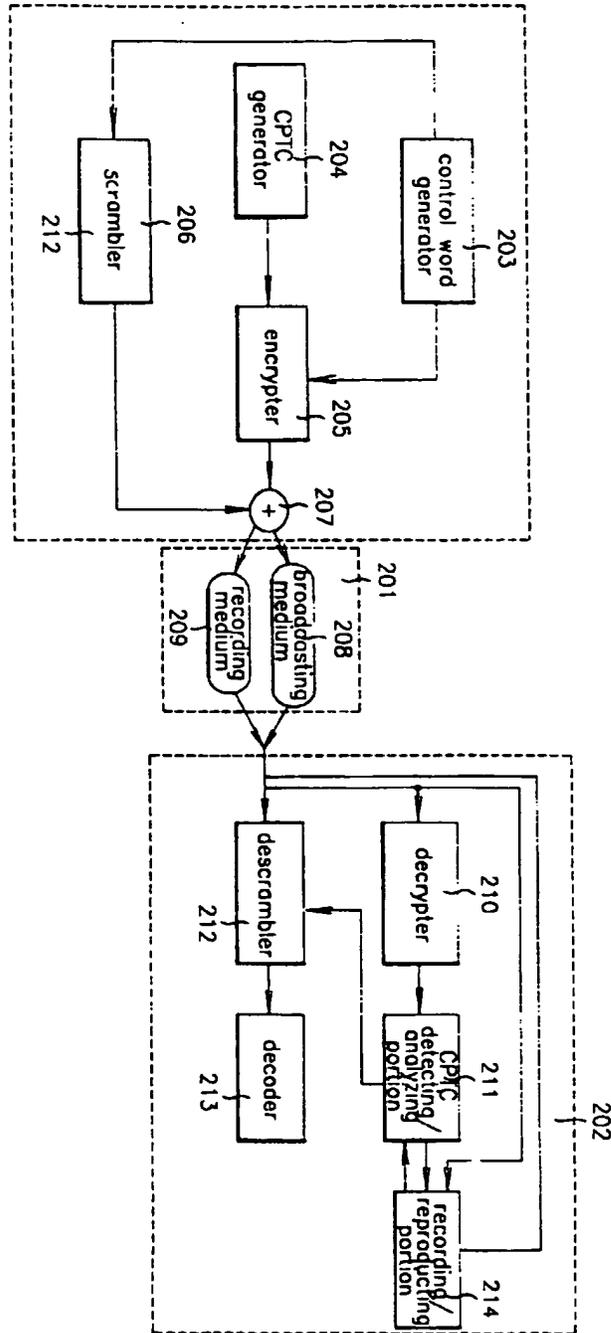
ID ₁	EMM ₁
ID ₂	EMM ₂
ID ₃	EMM ₃
⋮	⋮
ID _n	EMM _n

FIG. 14

recording/reproduction state	ID	reproduction number
------------------------------	----	---------------------

FIG. 15





F I G. 16

FIG. 17a

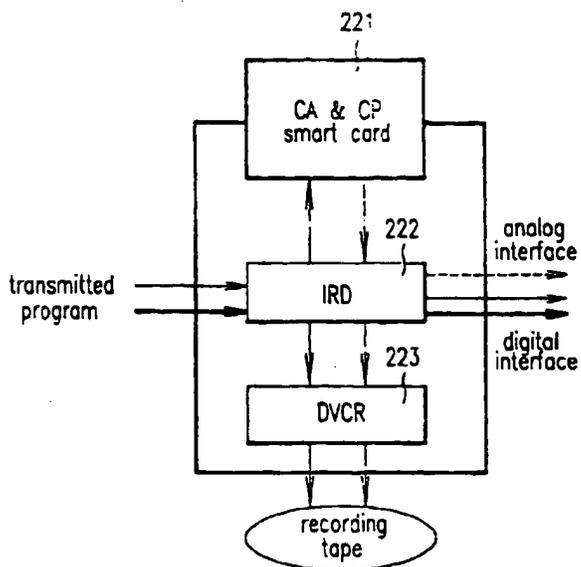


FIG. 17b

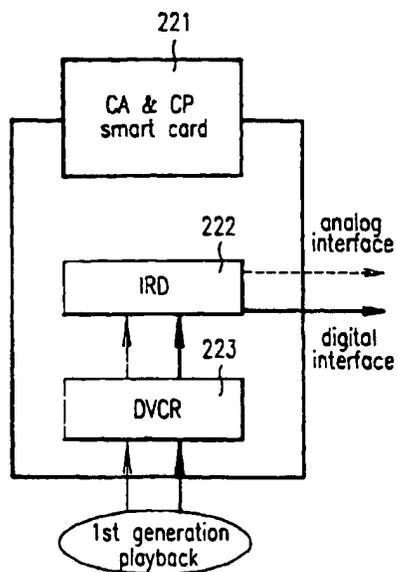


FIG. 18

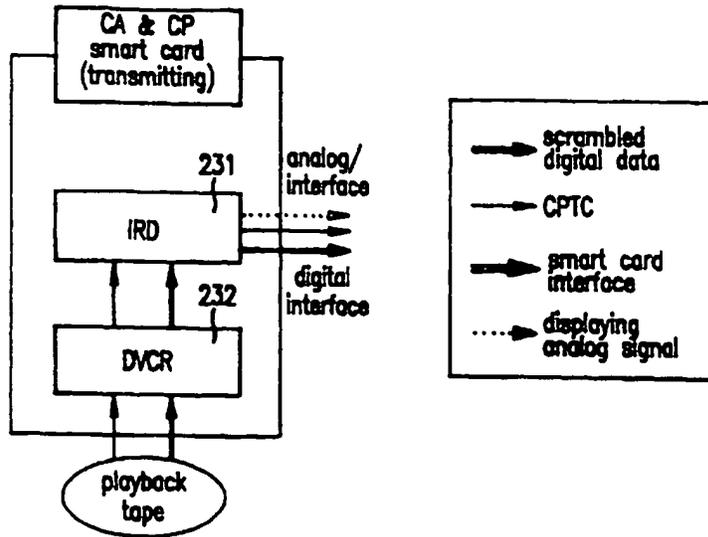


FIG. 19

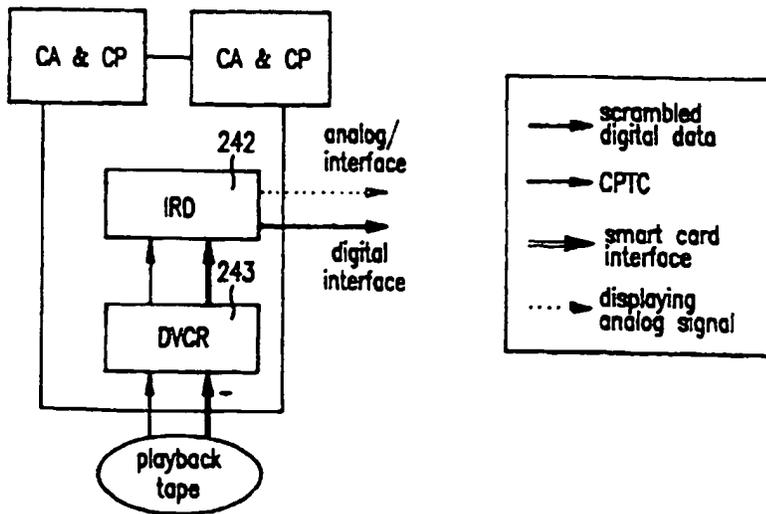


FIG. 20

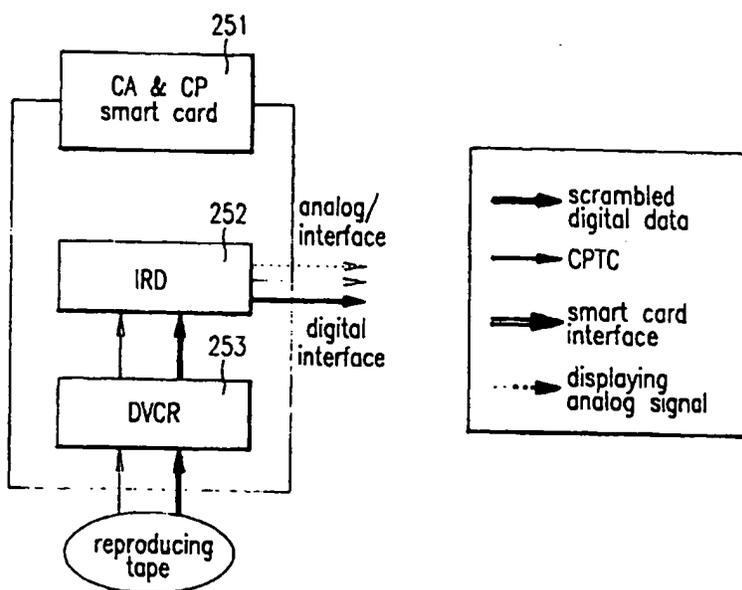


FIG. 21

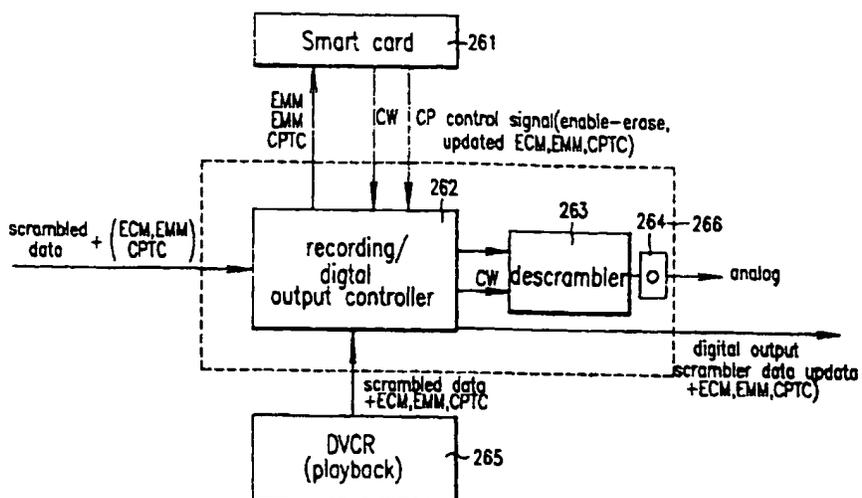


FIG. 22

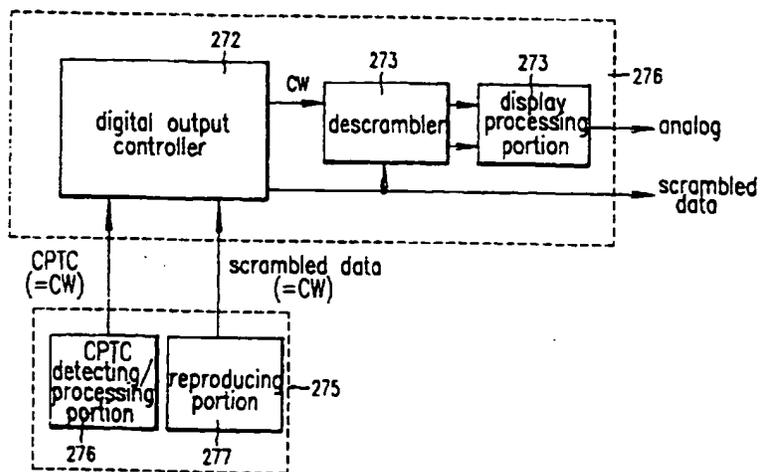


FIG. 23

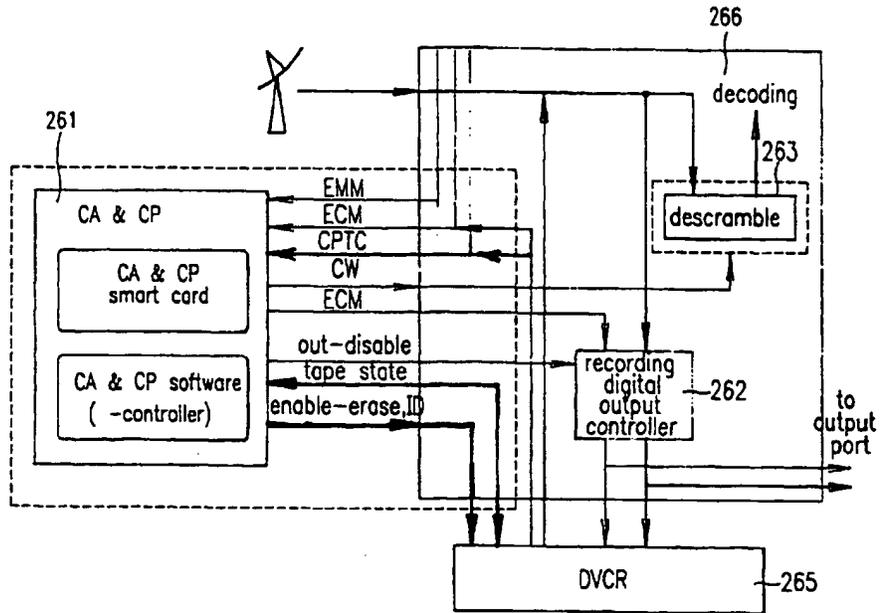


FIG. 24

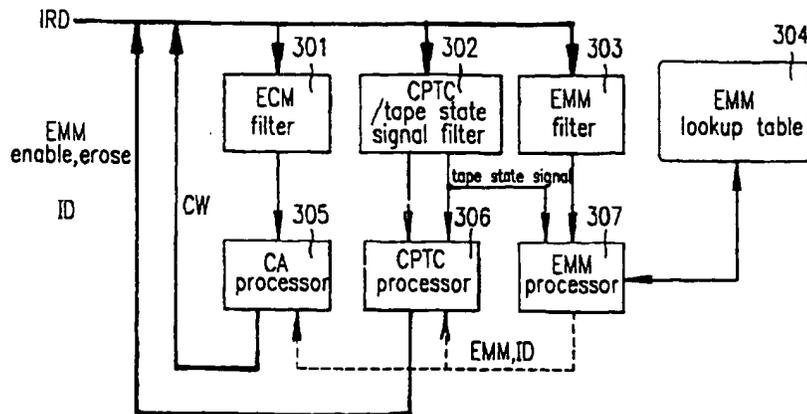


FIG. 25

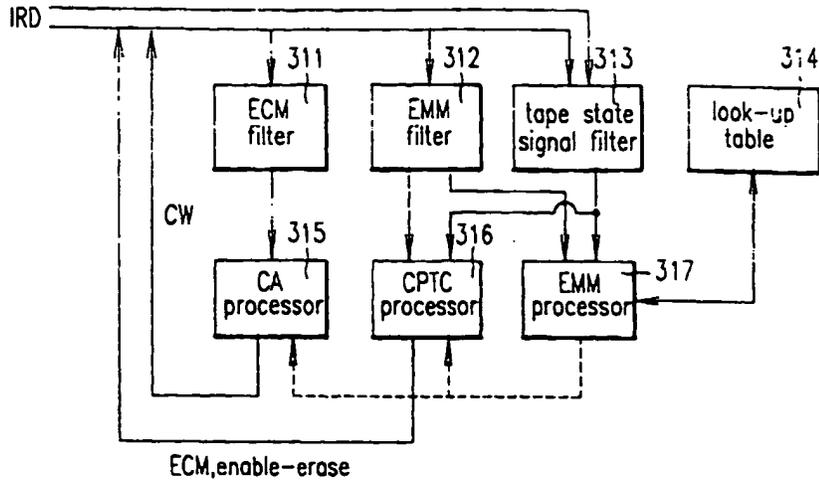
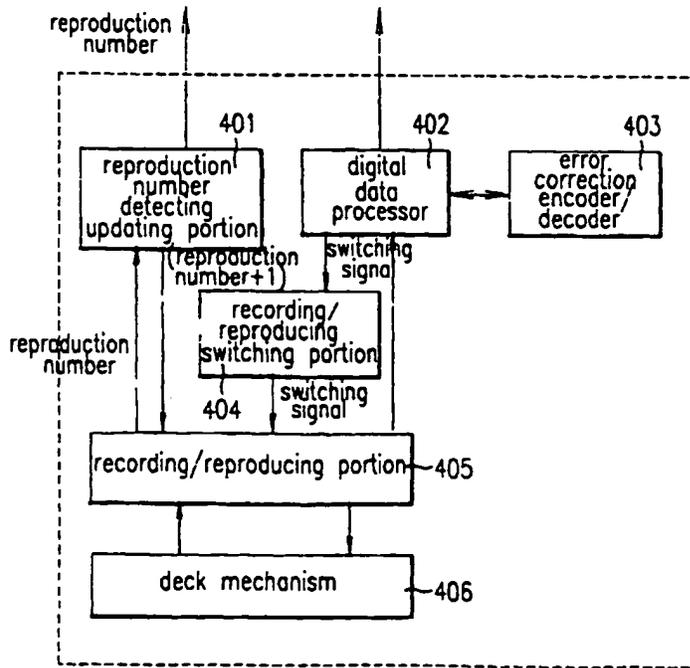


FIG. 26





(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication: 14.01.1998 Bulletin 1998/03 (51) Int Cl.⁶: G06F 17/60

(21) Application number: 97304946.3

(22) Date of filing: 07.07.1997

(84) Designated Contracting States: AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE	(72) Inventor: Kanno, Kazuhiro Koriyama-shi, Fukushima, 963-02 (JP)
(30) Priority: 08.07.1996 JP 178130/96 21.05.1997 JP 130626/97	(74) Representative: Cross, Rupert Edward Blount et al BOULT WADE TENNANT, 27 Furnival Street London EC4A 1PQ (GB)
(71) Applicant: Murakoshi, Hiromasa Koriyama-shi, Fukushima, 963 (JP)	

(54) **Software management system and method**

(57) An operation management system for managing the operation of a managed software product. When a management target function is executed, reference is made to a battery value and, if the value is zero or greater, the function is allowed to be executed. The battery

value is decremented as the function is executed. A charge value is supplied on a charge disk, such as a floppy disk, to allow the user to increase the battery value and to extend the usage period of the managed software product. The charge value may be supplied over a communication line.

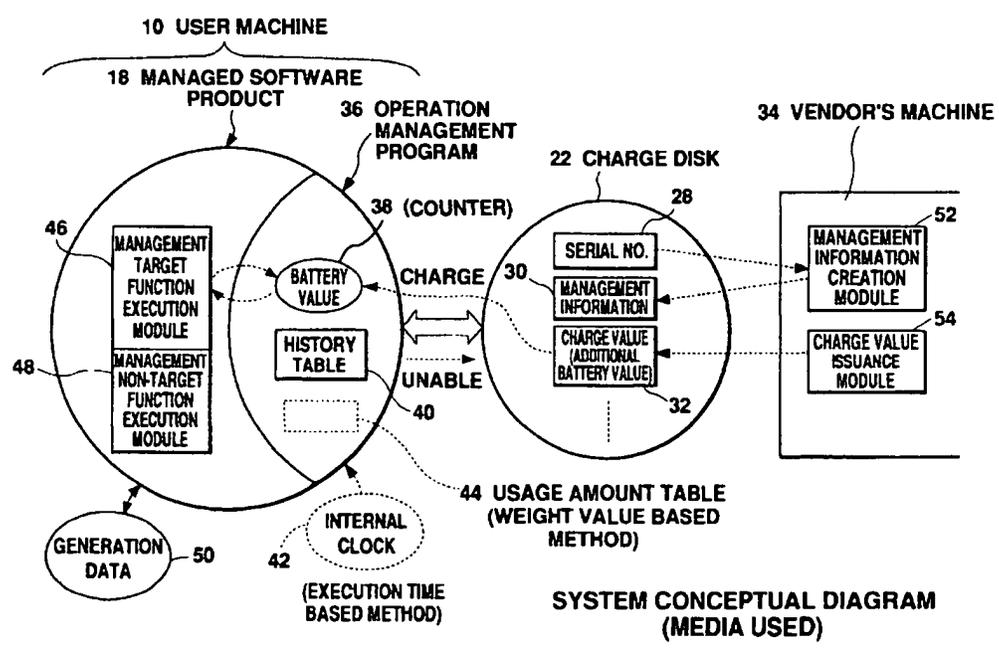


Fig. 3

EP 0 818 748 A2

Description**BACKGROUND OF THE INVENTION**

Field of the Invention

This invention relates to an operation management system and an operation management method, and more particularly to software operation management or execution management.

Description of the Related Art

As computers and computer use become more common, more advanced technology is introduced and a variety of software products are developed for use in various fields. However, in many cases, the user finds it difficult to select a product from among a variety of software products that seem to meet the user's requirements; often, the user cannot find the best tool for his needs.

To reduce such a risk, a service has been available that supplies the user with a trial-use software product free of charge. However, most of these trial-use software products contain only function descriptions or provide the user with limited functions (e.g., save function and/or output function is/are not included). This makes it difficult for the user to evaluate the actual product (all the functions) correctly.

A sales system which charges the user according to how long the user actually uses a software product (including a trial use) would allow him to buy the product anytime he wants, to fully evaluate the product, and to precisely determine the requirements for continued use (including payment for it). Many users would find this type of sales system appealing and economical.

In Japanese Patent Laid-Open Publication No. Sho 59-41061 and Japanese Patent Laid-Open Publication No. Sho 63-153633, a system is disclosed that automatically prevents a program from being used when the usage count reaches a specified value. In Japanese Patent Laid-Open Publication No. Hei 1-147622 a system is disclosed which accumulates program execution time (total program execution time) and prevents the program from being used when the accumulation time reaches a specified amount. However, these systems do not disclose means for extending the program usage period. Japanese Patent Laid-Open Publication No. Hei 5-134949 discloses a system in which a program and expiry of the program are downloaded from a host computer to a user computer via a communication line. Also disclosed is a system in which a new expiry of the program is downloaded from the host computer to the user computer in order to update the expiry. However, the system only measures the execution time taken for executing the entire program, and does not include any means for changing the expiry on the user computer.

In Japanese Patent Laid-Open Publication No. Hei

7-234785, a system is disclosed that relates to a software rental system. This system connects a computer in a rental company to a user computer on which a rental software product is running over a communication line. When the time elapsed from the rental start time reaches the rental limit time, the system makes the program unavailable for use. (For example, the program is deleted.) To allow the user to update the rental period, the rental company sends a rental period extension program to the user's computer over a communication line. The user runs this program to extend the rental period of the program. A drawback of this system is that the user must pay for the software product regardless of whether the user has used it frequently or not. This means that the amount of money the user has to pay depends, not on how often he has used it, but on how long he has used it.

In Japanese Patent Laid-Open Publication No. Hei 7-244585, a system is disclosed that manages the program usage period. This system assigns a usage limit date to a program and, when the current date becomes greater than the limit date, the program product is made unavailable. To extend the usage limit date, the system reads update limit data from a recording medium containing that data and re-assigns a usage limit date based on the update limit data. This system is not reasonable because the amount of money the user has to pay does not depend on whether or not the user actually uses the program.

For example, during execution of a Computer Aided Design (CAD) software product, the user often spends much time thinking without entering data. In the system disclosed by the above mentioned Japanese Patent Laid-Open Publication No. Hei 7-234785 or Japanese Patent Laid-Open Publication No. Hei 7-244585, the user must pay for this thinking time. This places unwanted pressure on the user, especially when he must think carefully during program execution.

SUMMARY OF THE INVENTION

The present invention seeks to solve the problems associated with the art described above. In view of the foregoing, it is an object of the present invention to provide an operation management system and method which reasonably manage the operation of a managed software product.

It is another object of the present invention to provide an operation management system and method which levy a charge according to the actual usage amount of the managed software product (or the amount of the result generated by the managed software product).

It is still another object of the present invention to provide an operation management system and method which manage the operation according to the property of each function of the managed software product.

(1) To achieve the above objects, an operation manage-

ment system for managing the operation of a managed software product according to the present invention comprises: battery value management means for decrementing a battery value according to the operation amount of the managed software product; operation limit means for limiting the operation of the managed software product when the battery value has decreased to a specified limit value; and charge means for adding a charge value to the current battery value when the charge value is entered from external means.

The "battery value" mentioned above is a "virtual battery" which drives a managed software product. This battery value is preferably the value of a counter.

The battery value management means decrement the battery value according to the operation amount of the managed software product. When the battery value has reached a specified limit value (for example, 0), the operation limit means limit all of or a part of the operation of the managed software product. Upon receiving a charge value (additional battery value) from the external means, the charge means add the received value to the current battery value, thus extending the operation period. That is, the battery value is incremented, just as a battery is charged, to allow the continued use of the managed software product.

The managed software product described above is preferably a packaged application software program including a CAD program, game program, video program, language processor, music program, communication program, or a measurement program.

The battery value management means, operation management means, and charge means described above should be implemented preferably as software programs (management software programs) that run on a computer. The managed software product and the management software product may be separate, or the whole or a part of the management software product may be included in the managed software product.

A system according to the present invention is implemented on a general-purpose computer or special-purpose computer having such peripheral units as a disk drive, display, and input unit. The external means described above include recording media such as a magnetic disk or an optical disk and other host computers connected over a network.

(2) An operation management system according to the present invention may be applied to an application software product sales system. The following explains an example:

A vendor sells an application software product containing the operation management program according to the present invention. The operation management program has a battery value defined as the initial value. In addition to this product, the vendor sells recording media containing charge values (e.g., floppy disk (FD)). In this case, it is desirable that a variety of recording media, each containing a unique charge value, be supplied.

On the other hand, a user who bought the application software product may use the product until the battery value reaches zero. This allows the user to fully evaluate and examine the product. A user who wants to use the product after the battery value becomes zero must buy a recording medium containing a charge value to charge the battery. This enables him to add a charge value to the battery value and to use the product continuously.

If the specifications of the application software product do not satisfy the user's request, the user does not buy the recording medium. This prevents additional charges and reduces the cost to the user.

Considering an increase in the sales profit in recording media that will be produced in the future, a combination of a managed software product and the operation management program will lower prices significantly. The operation management system according to the present invention will increase the profits of both the user and the vendor, making it possible to build a very reasonable, economical system.

(3) In a preferred embodiment of the present invention, the battery value management means calculate the operation amount of each function of the managed software product, and subtracts a value corresponding to the operation amount from the battery value.

A continuous decrease in the battery value during execution of a managed software product, as in a conventional system, decrements the value even when the user is idle (input wait time), which places pressure on the user.

Calculating the operation amount of each function during execution of a managed software product, as in a system according to the present invention, decreases the battery value only when the managed software product is actually used, enabling the user to do operation without having to worry about time elapsed while thinking.

(4) In a preferred embodiment of the present invention, function category determination means are also available which determine if an execution instruction from the user activates a management target function or a management non-target function. And, the battery value management means decrement the battery value only when the management target function is executed.

For example, with the data generation function defined as a management target function and with other functions as management non-target functions, a cost can be levied only when new data are generated.

(5) In a preferred embodiment of the present invention, the battery value management means have a weight table containing an operation amount weight value for each of the management target functions. When any of the management target functions is executed, the battery value management means decrement the battery value by the weight value corresponding to the management target function.

In a preferred embodiment of the present invention,

the battery value management means measure the execution time of each of the management target functions and decrement the battery value by the value corresponding to the execution time.

This weight value system is able to calculate the operation amount regardless of the computer speed, which may differ among computers. In addition, by measuring time in this manner, the execution time is directly monitored and therefore the operation amount becomes proportional to the CPU load.

(6) In a preferred embodiment of the present invention, the operation limit means prevent only the management target functions from being executed when the battery value has decreased to a specified limit value; management non-target functions are executed.

For example, forcing a game program used at home to terminate when the battery value has reached a specified value does not cause a serious problem.

However, for a CAD program used in an office, forced termination when the battery value has reached a specified value may make already-produced data unavailable, possibly interrupting a job. Therefore, considering user's advantage and convenience, the embodiment keeps some functions operable even when the battery value has reached a specified value.

(7) A preferred embodiment of the present invention has remainder warning means for issuing a remainder warning message when the battery value has decremented to a specified warning value because a sudden inoperable condition in the managed software product without prior notice may cause the user unexpected damage. The remainder warning means alert the user to that condition before it occurs. In other words, the warning message prompts the user to determine whether to charge the battery value.

A preferred embodiment of the present invention has remainder display means for displaying the battery value on the screen during execution of the managed software product. This remainder display information keeps the user informed of the amount by which the managed software product will be able to continue operation without being charged.

It is also possible to program the system so that, upon detecting that the battery value has been charged to a specified value, the system can automatically disable operation management through the battery value to allow the user to use the product indefinitely.

(8) To achieve the above objects, a method for managing the operation of a managed software product according to the present invention comprises: a count value management step for changing a count value according to the operation amount of the managed software product; an operation limit step for limiting the operation of the managed software product when the count value has reached a specified limit value; and a charge step for charging the current count value or the limit value when a charge value is entered from external means.

The above count value is incremented or decre-

mented according to the operation amount of the managed software product. When the count value is incremented, a charge value is added to the limit value; when the count value is decremented, a charge value is added to the current count value. In either case, the usage period is extended by charging the battery value.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram showing a user machine used in the operation management system according to the present invention.

Fig. 2 is a diagram showing the data structure of a charge disk.

Fig. 3 is a diagram showing the concept of the operation management system according to the present invention.

Fig. 4 is a diagram showing an example of the history table.

Fig. 5 is a diagram showing an example of the usage amount table.

Fig. 6 is a flowchart showing the processing of the system when a management target function is executed in the execution time based method.

Fig. 7 is a flowchart showing the processing of the system when a management target function is executed in the weight value based method.

Fig. 8 is a flowchart showing the charge disk read processing.

Fig. 9 is a flowchart showing the charge processing.

Fig. 10 is a diagram showing a user machine used in another embodiment.

Fig. 11 is a diagram showing the structure of data sent from the host machine to a user machine.

Fig. 12 is a diagram showing the concept of the system in another embodiment.

Fig. 13 is a diagram showing an example of the user registration table.

Fig. 14 is a flowchart showing the operation of the user machine and a user machine in another embodiment.

Fig. 15 is a diagram showing another configuration of the system.

Fig. 16 is a diagram showing an example of an application according to the present invention.

Fig. 17 is a flowchart showing the function category determination processing.

DESCRIPTION OF PREFERRED EMBODIMENTS

Fig. 1 shows a user machine 10. This user machine 10 is a computer which executes various types of application programs under control of the operation system (OS). The user machine 10 is composed of a system unit 12, display 14, keyboard (not shown in the figure), output unit (not shown in the figure) such as a printer or plotter, and so forth. The system unit 12 contains a CD-ROM disk drive 16 which accesses a CD-ROM and

reads data from it and a floppy disk drive 20 which accesses a floppy disk (FD) and reads data from it.

The CD-ROM shown in Fig. 1 contains a managed software product 18. In this embodiment, the managed software product 18, such as a CAD software product, has an operation management program built in. The operation management program, designed for managing the operation of the managed software product 18, manages the operation using a "battery value" which will be described below. In the example shown in Fig. 1, the managed software product 18 is installed from the CD-ROM to the user machine 10; it may be installed from any other recording medium or via a communication line.

A charge disk 22, containing specified data (including a charge value) on a floppy disk, functions as a battery value charger. Inserting this charge disk 22 into the floppy disk drive 20 causes a charge value to be read and enables the user to extend the allowable operation period of the managed software product 18. In this embodiment, several charge disks 22, each containing a unique charge value, are supplied to allow the user to select or buy a desired charge disk 22 to add a desired charge value to the battery value.

The managed software product 18 and the charge disk 22 are usually supplied from the same vendor. In this embodiment, the managed software product 18 includes the operation management program. Of course, the managed software product 18 and the operation management program may be separately loaded into the user machine 10.

In Fig. 1, the display 14 has a remainder information area 24 where remainder information is displayed and a remainder warning area 26 where a warning message is displayed when the remainder drops below the specified amount. These areas will be described later.

Fig. 2 shows the data structure of the charge disk 22. As shown in Fig. 2, the charge disk 22 contains a serial number 28, management information 30, and charge value (additional battery value) 32. The serial number 28 is a unique identification number that is assigned when the floppy disk is formatted. Usually, this number is not copied when the disk is copied. The management information 30 is created when the serial number 28 is encrypted. This management information 30 is copied when the disk is copied. Therefore, when the disk is copied illegally, the serial number 28 and the management information 30 do not match, thereby making it easy to determine that the disk is copied illegally. Of course, any other conventional security system may also be used instead of this method.

The charge value 32 is an additional charge value to be added to the battery value that is decremented as the user uses the managed software product 18. Charging the battery value with this charge value enables the user to extend the usage period.

When the battery value is managed in the "execution time based method" in which the battery value is

decremented by the execution time of each function, an additional time is recorded as the charge value 32. On the other hand, when the battery value is managed in the "weight value based method" in which the battery value is decremented by the weight value of each function, the additional value is recorded as the charge value 32. These methods will be described in more detail later.

Although a floppy disk is used as the charge disk 22 in the embodiment shown in Fig. 1, other types of recording media may also be used. Also, as shown in another embodiment that will be explained later, a charge value may be sent over a communication line.

Fig. 3 shows the concept of the operation management system which uses the charge disk 22. The system is composed primarily of the user machine 10, charge disk 22, and vendor's machine 34. In this embodiment, the managed software product 18 including the operation management program 36 is installed in the user machine 10.

The charge disk 22 is generated on the vendor's machine 34 owned by the vendor which sold the managed software product 18. More specifically, the vendor's machine 34 has two software modules: the management information creation module 52 and the charge value issuance module 54. The management information creation module 52 encrypts the serial number 28 recorded on the charge disk 22, and writes the resulting management information 30 back onto the charge disk 22. Note that the operation management program 36, which contains the encryption condition or the decryption condition, can check whether or not the serial number 28 agrees with the management information 30. The charge value issuance module 54 records the charge value 32, which has been set by the vendor, onto the charge disk 22. In the execution time based method, the charge value 32 is recorded, for example, as 100 hours, 200 hours, or 500 hours. Note that the operation management program 36 contains an initial battery value (for example, 100 hours).

The operation management program 36 has a counter 38 which decrements the battery value (battery value management function). In this embodiment, the operation management program 36 decrements the counter 38 each time a "management target function" provided by the managed software product 18 is executed. When the battery value, i.e., the counter value, has decremented to the limit value of 0, the operation management program 36 prevents management target functions from being executed. That is, in this embodiment, when the battery value has reached a specified limit value, the execution of the managed software product 18 is limited and, when the battery value is charged with the charge value 32 contained on the charge disk 22, the charge value is added to the battery value and the resulting value is used as a new battery value. The usage period of the managed software product 18 is thus extended.

A history table 40 managed by the operation man-

agement program 36 contains history information on charge values recorded on the charge disk 22. Fig. 4 shows an example. As shown in Fig. 4, the history table 40 is composed of three columns: FD serial number column 40A, charge data/time column 40B, and charge value column 40C. The table may have other columns as necessary.

Referring to Fig. 3 again, the following explains how the battery value is managed. When the battery value is managed in the "execution time based method" described above, the execution time of each management target function, measured based on the internal clock 42, is subtracted from the battery value. On the other hand, when the "weight value based method" described above is used, the battery value is managed based on the usage amount table 44. Fig. 5 shows an example of the usage amount table 44. In this embodiment, the table contains entries, each consisting of a function name 44A and the corresponding usage amount 44B. It should be noted that each usage amount is used as a weight value. For example, a weight value is pre-defined according to the processing time of each function. Therefore, when a management target function is executed, the corresponding usage amount (weight value) is subtracted from the battery value.

The managed software product 18 shown in Fig. 3 has many user interface programs as well as many internal functions and common functions used by the programs. These functions are classified roughly into two: management target functions and management non-target functions. Whenever the managed software product 18 attempts to execute a management target function, the operation management program 36 references the battery value and, when it is zero or greater, allows the managed software product 18 to execute that function. When the managed software product 18 attempts to execute a management non-target function, the operation management program 36 does not check the battery value. For example, when input/output function for processing generated data 50 from the managed software product 18 is defined as a management non-target function, the input/output processing is always executed on the generated data 50, even if the usage period of the managed software product 18 has expired. This ensures that the generated data 50 are always processed, thus protecting user assets. Examples of management non-target functions include the data display function, data print function, and data plotter output function.

Management target functions include the data generation function. For example, when the managed software product is a CAD software product, the data generation function includes the straight-line drawing function, curved-line drawing function, circle drawing function, area fill-in function, area hatching function, and character insertion function.

Fig. 3 conceptually shows management target function execution module 46 which executes management

target functions and management non-target function execution module 48 which executes management non-target functions. In this embodiment, the battery value is decremented only when a management target function is activated. Note that the battery may be decremented when both a management target function and a management non-target function are activated.

In addition to the data described above, the charge disk 22 may contain other types of data. For example, it may contain the name of the managed software product 18 which accepts a charge value. In this case, the name of the managed software product 18 is used as follows. When the charge disk 22 is read, the operation management program 36 checks whether or not the name of the managed software recorded on the charge disk 22 matches that of the managed software product 18 installed in the user machine 10 and, only when they match, accepts the charge value 32.

The battery value described above is stored on the hard disk and then copied into the computer's RAM. The battery value in the RAM is decremented whenever a management target function is executed. Also, at an interval or as necessary, the battery value in the RAM replaces the battery value on the hard disk. This means that, even when the computer fails, the battery value is not erased. The battery value may also be maintained in some other way.

Fig. 17 is a flowchart showing how the operation management program operates when it accepts an instruction requesting the execution of a managed software product function. The following explains this processing in more detail.

Upon receiving from a user an instruction requesting the execution of a function of the managed software product while the managed software product is in execution (S601), the operation management program checks whether the requested function is a management target function or a management non-target function (S602). When the function is a management target function (S603), the operation management program performs the processing shown in Fig. 6 or Fig. 7 (S604). When the function is a management non-target function (S603), the program executes the function immediately (S605). This processing is repeated whenever an execution instruction is received.

Next, referring to Fig. 3, the execution of a management target function in the execution time based method is explained with the use of Fig. 6.

When the user requests the execution of a management target function while the managed software product 18 shown in Fig. 3 is in execution, the routine shown in Fig. 6 is started. First, the management target function execution module 46 or the operation management program 36 reads the battery value to check if it is greater than zero. If the battery value is zero or less, the routine is terminated. That is, the requested management target function cannot be started. Note that a management non-target function is started even if the battery value is

zero.

In S102, the routine gets the start time from the internal clock 42 before starting the requested management target function and, in S103, starts the management target function. In S104, the routine gets the end time from the internal clock 42 and, in S105, subtracts the start time from the end time to calculate the processing time (execution time) of the processing executed in S103.

In S106, the routine subtracts the processing time calculated in S105 from the battery value. In S107, the routine checks if the resulting battery value is equal to or less than the warning value and, if so, displays a message in the remainder warning area 26 shown in Fig. 1. If the resulting battery value is greater than the warning value, the routine does not display the message. As shown in Fig. 1, the remainder information area 24 is displayed during execution of the managed software product 18 (see Fig. 1) to allow the user to check the remaining amount. This helps the user determine how long he can execute the managed software product 18.

Fig. 7 shows the processing of a management target function in the weight value based method.

When the execution of a management target function is requested as described above, the routine references the battery value in S201 to check if it is equal to or greater than 0. If it is, the routine executes the requested management target function in S202 and, in S203, references the usage amount table 44 shown in Fig. 5 to find the usage amount (weight value) of the executed management target function. Then, in S204, the routine subtracts the processing amount found in S203 from the battery value to find a new battery value. In S205, the routine checks if the battery value is less than the warning value and, if so, displays a message in the remainder warning area 26 in S206.

The "execution time based method" shown in Fig. 6 allows the user to manage operation using a physical amount that is easy to understand. In addition, the user can manage operation in a relatively simple configuration. On the other hand, the "weight value based method" shown in Fig. 7 gives the user the same result regardless of the CPU speed of the user's machine.

Next, referring to Fig. 3, the charge disk 22 read processing is explained with the use of Fig. 8.

This processing is started when the charge disk 22 is inserted into the floppy disk drive 20 as shown in Fig. 1. The routine reads the serial number in S301, and the management information in S302, both from the charge disk 22. In S303, the routine encrypts the serial number according to the encryption condition, or decrypts the management information according to the decryption condition, and compares the serial number with the management information. This comparison determines whether or not the charge disk 22 is legal. For example, when the disk is illegally copied, the management information 30 is copied, but the serial number 28 is not copied but replaced. This results in a mismatch between the

serial number 28 and the management information 30, thereby making it possible to find an illegal copy.

In S304, the routine checks if the charge disk 22 is valid and, if it is not valid, terminates processing in S308. If it is valid, the routine references the history table 40, containing past charge history data, in S305 to check the validity of the charge value 32 recorded on the charge disk 22. To do so, the routine first checks to see if the serial number 28 of the charge disk 22 is in the history table 40. If the serial number is found, the routine takes the following steps to check if the charge value 32 recorded on the charge disk 22 is valid. The routine finds the charge value initially recorded on the charge disk 22 and, from that initial value, subtracts the actual charge value to find the remainder. The next time the battery value is charged, the routine compares the remainder with the charge value currently recorded on the charge disk. If the charge value on the charge disk 22 is greater than the remainder, the routine determines in S306 that the charge disk is not valid and terminates processing in S308. If the routine finds that the charge value 32 on the charge disk 22 is valid, it performs the charge processing, shown in Fig. 9, in S307.

Fig. 9 shows an example of charge processing. In S401, the routine references the counter 38 to read the current battery value and, in S402, reads the charge value from the charge disk 22. In S403, the routine asks the user to type an actual charge value that does not exceed the charge value 32 recorded on the charge disk 22. The user types the charge value, for example, from the keyboard. In S404, the routine checks that the specified charge value is less than the charge value on the charge disk 22. If the specified charge value is greater than the charge value on the charge disk 22, the routine asks the user to retype the charge value.

In S405, the routine adds the specified charge value to the battery value, thus charging the battery value. In S406, the routine subtracts the specified charge value from the initial charge value and writes the resulting value on the charge disk 22 as a new charge value 32. If the initial charge value 32 is exhausted, the routine writes the value of 0 on the charge disk 22 to virtually erase the charge value. The value of 0 prevents the charge disk 22 from being re-used. In S407, a record relating to the charge processing is added to the history table 40.

In the above embodiment, the user specifies an actual charge value. Instead of having the user specify a value, a pre-defined charge value may be added to the battery value at that time.

Fig. 10 shows another embodiment according to the present invention. In the embodiment described above, the battery value is charged using a recording medium. In this embodiment, the battery value is charged via a communication line 60. For the same components as those used in the above embodiment, the same numbers are assigned and their descriptions are omitted.

The user machine 10 in Fig. 10 is connected to the

host machine 62 via the communication line 60. From this host machine 62, send data 64 shown in Fig. 11 are sent to the user machine 10 to charge the battery value.

In Fig. 11, address information 68 specifies the address of the user machine 10. Management information 70 is created by encrypting the serial number on the recording medium containing the managed software product 18. A charge value 72, a value to be added to the battery value as with the above embodiment, is an additional period of time in the execution time based method, and is an additional amount in the weight value based method.

Fig. 12 illustrates the system concept of this embodiment.

As described above, the user machine 10 is connected to the host machine 62 via the communication line 60. That is, this host machine 62 is connected to each of a number of user machines 10 for integrated operation management. This host machine 62 has a management information creation module 76, charge value issuance module 78, user registration table 80, and billing module 82. The management information creation module 76 creates the management information 70 shown in Fig. 11, and the charge value issuance module 78 issues a charge value 72 in response to a request from the user machine 10. As shown in Fig. 13, the user registration table 80 is composed primarily of the user ID column 80A, user name column 80B, and request charge value column 80C. The billing module 82 references the user registration table 80 to automatically issue a bill for a requested amount whenever a charge value is issued, or at some specified interval.

Next, referring to Fig. 12, the operation of this embodiment is explained with the use of Fig. 14. The operation of the user machine 10 is shown in the left side of Fig. 14, while that of the host machine 62 is shown on the right.

First, in S501 and S502, the user machine 10 is connected to the host machine 62 via a communication line. In S503, the user machine 10 generates a request for a charge value that will be sent to the host machine 62. In this case, the request contains at least the serial number of the CD-ROM containing the managed software product 18 and information on the charge value. In S504, the user machine sends the request to the host machine and, in S505, the host machine receives the request.

In S506, the host machine checks the user registration table 80. If the host machine finds, in S507, that the requesting user is registered in the host machine 62, the management information creation module 76 creates management information based on the serial number in S508, and the charge value issuance module 78 generates a charge value in response to the request from the user. In S509, the host machine 62 sends the management information and the charge value to the user machine 10 as the send data 64 shown in Fig. 11. In S510, the user machine 10 receives the send data 64. In S511 and S512, the user machine 10 and the host machine

62 are disconnected.

In S513, the operation management program 36 compares the serial number 74 with the management information 70 to check to see if the data received by the user machine 10 are valid. This prevents the user from illegally charging the battery value. If it is found in S514 that the send data are valid, the charge processing is performed in S515. This charge processing is the same as that in Fig. 9.

As shown in Fig. 12, this embodiment may also use the execution time based method or the weight value based method in order to manage the battery value.

Although the battery value is charged over a communication line such as a telephone line in the above embodiment, it may also be charged over a communication satellite (satellite line).

In the above embodiments, the operation management program 36 is included in the managed software product 18. Of course, an external program can manage the operation of the managed software product 18. Fig. 15 shows the concept of such an embodiment.

As shown in Fig. 15, the operation system (OS) 83 is located between the hardware 81 and each of application programs 84, 86, and 88. The operation management program 36 according to the present invention may be located between the operation system 83 and the application program 84.

Operation management program 36 therefore functions as an interface program. Messages are exchanged between the operation management program 36 and the application program 84 according to some specific rule. Messages are also exchanged between the operation management program 36 and the operation system 83 according to a specific rule.

To execute a management target function in this configuration, the operation management program 36 references the battery value when it receives an execution request from the application program 84. If the battery value is not zero, the operation management program 36 sends an instruction to the operation system 83 while simultaneously decrementing the battery value by a value corresponding to the function. If the battery value is zero, the operation management program 36 sends a message back to the application program 84, indicating that the instruction cannot be executed.

To execute a management non-target function, the operation management program 36 does not reference the battery value when it receives an execution request from the application program 84 but instead sends the instruction directly to the operation system 83.

The battery value is decremented as management target functions are executed. Charging the battery value allows the user to extend the usage period of the application program 84, which may be supplied separately from the application program 84.

In the above embodiments, one operation management program manages one operation management program. It is also possible for one operation manage-

ment program to manage several application programs.

Fig. 16 shows an application of the present invention. The system shown in Fig. 16 is composed of one host machine 90 and several user machines 92. Within each user machine 92 are a managed software product 18 and the operation management program 36, which, in turn, contains the counter 38 where the battery value to be decremented is stored. In other words, the operation of the managed software product 18 is controlled by the value stored in the counter 38. To execute the managed software product 18 in this system, it is necessary to insert a battery disk 96 into the user machine 92 and to move the battery value from the battery disk 96 into the counter 38. The battery value is decremented as the operation of the managed software product 18 proceeds. When the user finishes the managed software product 18, a sequence of operations are executed to move the current counter value from the counter 38 to the battery disk 96. This initializes the counter 38 to zero just as it was before the battery disk 96 was inserted.

The host machine 90 has several disk drives into which a battery disk 96 is inserted to read the battery value that was returned to the battery disk 96. This host machine 90 is also used to charge the battery value on the battery disk 96.

Integrated management of the battery values on several battery disks 96 through the host machine 90 brings a benefit of integrally managing several managed software products 18.

This type of system may be used, for example, in a school or a business where many computers are installed. With an individual carrying his or her own portable battery disk 96, it is possible to check and control the software usage amount of each person. In this case, either the "execution time based method" or the "weight value based method" may be used.

Claims

1. An operation management system for managing the operation of a managed software product, comprising:

battery value management means for decrementing a battery value according to the operation amount of said managed software product;

operation limit means for limiting the operation of said managed software product when said battery value has decremented to a specified limit value; and

charge means for adding a charge value to the current battery value when the charge value is entered from external means.

2. An operation management system according to

claim 1, wherein said battery value management means find the operation amount for each execution of a function owned by said managed software product and subtract a value corresponding to said operation amount from said battery value.

3. An operation management system according to claim 2, further comprising:

function category determination means for determining if a function to which an execution instruction is issued is a management target function or a management non-target function, wherein said battery value management means decrement said battery value only when said management target function is executed.

4. An operation management system according to claim 3, wherein

said battery value management means has a weight table containing pairs of said management target function and a weight value representing said operation amount thereof, and said battery value management means subtract a weight value corresponding to said management target function from said battery value when said management target function is executed.

5. An operation management system according to claim 3, wherein, when said management target function is executed, said battery value management means measure the execution time and subtracts the execution time from said battery value.

6. An operation management system according to claim 3, wherein said operation limit means prevent said management target function from being executed but allows said management non-target function to be executed when said battery value has reached a limit value.

7. An operation management system according to claim 3, wherein said managed software product has a data generation function and a data output function and wherein said function category determination means determine said data generation function as said management target function and determine said data output function as said management non-target function.

8. An operation management system according to claim 1, further comprising remainder warning means for issuing a remainder warning when said battery value has decremented to a warning value.

9. An operation management system according to claim 1, further comprising remainder display

means for displaying said battery value during execution of said managed software product.

10. An operation management system for managing the operation of a managed software product, comprising:

battery value management means for decrementing a battery value according to the operation amount of said managed software product;

operation limit means for limiting the operation of said managed software product when said battery value has decremented to a specified limit value;

read means for reading a charge value from a recording medium containing the charge value thereon; and

charge means for adding said charge value to the current battery value.

11. An operation management system according to claim 10, further comprising erase means for erasing the charge value from said recording medium after said charge value is added.

12. An operation management system according to claim 10, further comprising:

specification means for allowing a user to specify an actual charge value by which the current battery value is to be actually charged, the actual charge value not exceeding the charge value recorded on said recording medium; and
rewrite means for rewriting the charge value on said recording medium with a remainder value after said actual charge value is added to the current battery value.

13. An operation management system according to claim 10, in which said recording medium contains not only said charge value, but also the identification number of the recording medium and management information generated through encryption of the identification number, said operation management system further comprising:

validity determination means for comparing said identification number with said management information considering the condition of said encryption to determine the validity of said recording medium.

14. An operation management system comprising:

a managed machine containing a managed software product; and

a managing machine connected to said managed machine with a communication line,

wherein

said managed machine comprises:

battery value management means for decrementing a battery value according to the operation amount of said managed software product;

operation limit means for limiting the operation of said managed software product when said battery value has decremented to a specified limit value;

charge value receive means for receiving a charge value from said managing machine; and
charge means for adding said charge value to the current battery value, and wherein

said managing machine comprises:

charge value send means for sending said charge value to said managed machine.

15. An operation management system according to claim 14, wherein said managed machine further comprises:

notification means for notifying said managing machine of the identification number of a portable recording medium initially containing said managed software product; and

validity determination means for comparing management information sent from said managing machine with said identification number to determine the validity of the recording medium; and wherein said managing machine further comprises:

management information creation means for creating said management information generated by encrypting said notified identification number and for sending the management information to said managed machine.

16. An operation management system comprising:

at least one managed machine containing a managed software product; and

a managing machine for managing the operation of said managed machine, wherein said managed machine comprises:

a counter containing a battery value changing according to the operation amount of said managed software product;

first charge means for reading a battery value from a portable recording medium to store the battery value into said counter; and

first return means for writing the current battery value on said recording medium, and wherein, said managing machine comprises:

second charge means for writing said battery value on said recording medium; and

second return means for reading said battery value from said recording medium.

17. An operation management method comprising:

a count value management step for changing
a count value according to the operation
amount of a managed software product; 5
an operation limit step for limiting the operation
of said managed software product when said
count value has reached a specified limit value;
and
a charge step for charging the current count val- 10
ue or said limit value when a charge value is
entered from external means.

a module for charging the current count value
or said limit value when a charge value is en-
tered from external means.

18. A medium containing a management software prod- 15
uct for managing the operation of a managed soft-
ware product, wherein said managed software
product and said management software product are
executed on computers, said management soft-
ware product comprising:

a module for changing a count value according
to the operation amount of said managed soft-
ware product; 20
a module for limiting the operation of said man-
aged software product when said count value 25
has reached a specified limit value; and
a module for charging the current count value
or said limit value when a charge value is en-
tered from external means. 30

19. A medium containing a charge value read by a man-
agement software product for use in managing the
operation of a managed software product, wherein
said managed software product and said manage-
ment software product are executed on computers, 35
said management software product comprising:

a module for changing a count value according
to the operation amount of said managed soft-
ware product; 40
a module for limiting the operation of said man-
aged software product when said count value
has reached a specified limit value; and
a module for charging the current count value 45
or said limit value when said charge value is
entered.

20. A computer system having an interface software
product between an operation system and at least
one application software product, wherein said in- 50
terface software product comprises:

a module for changing a count value according
to the operation amount of said application soft-
ware product; 55
a module for limiting the operation of said ap-
plication software product when said count val-
ue has reached a specified limit value; and

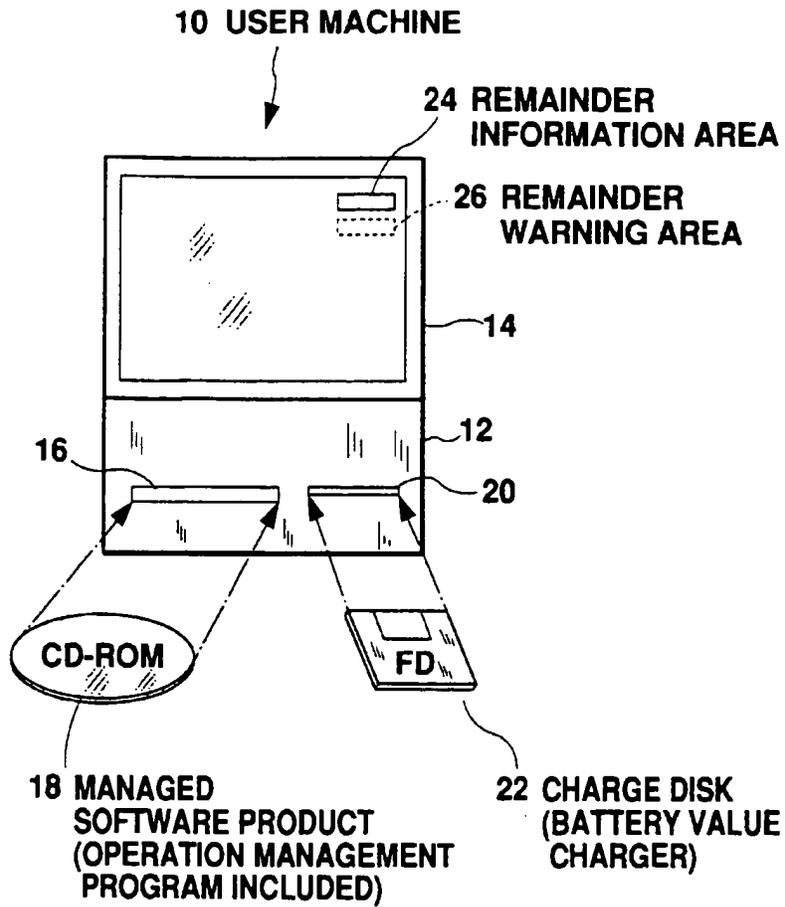


Fig. 1

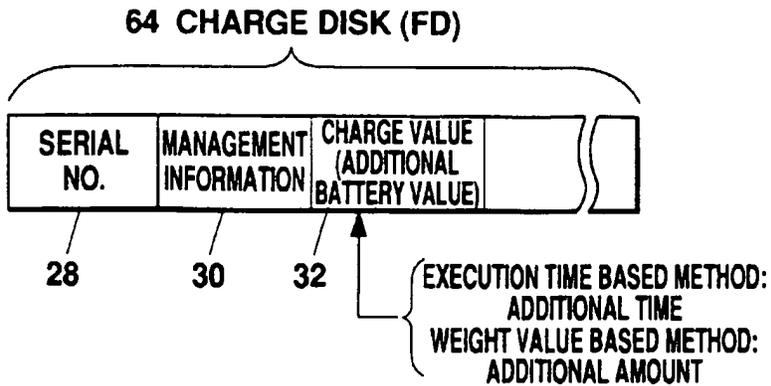


Fig. 2

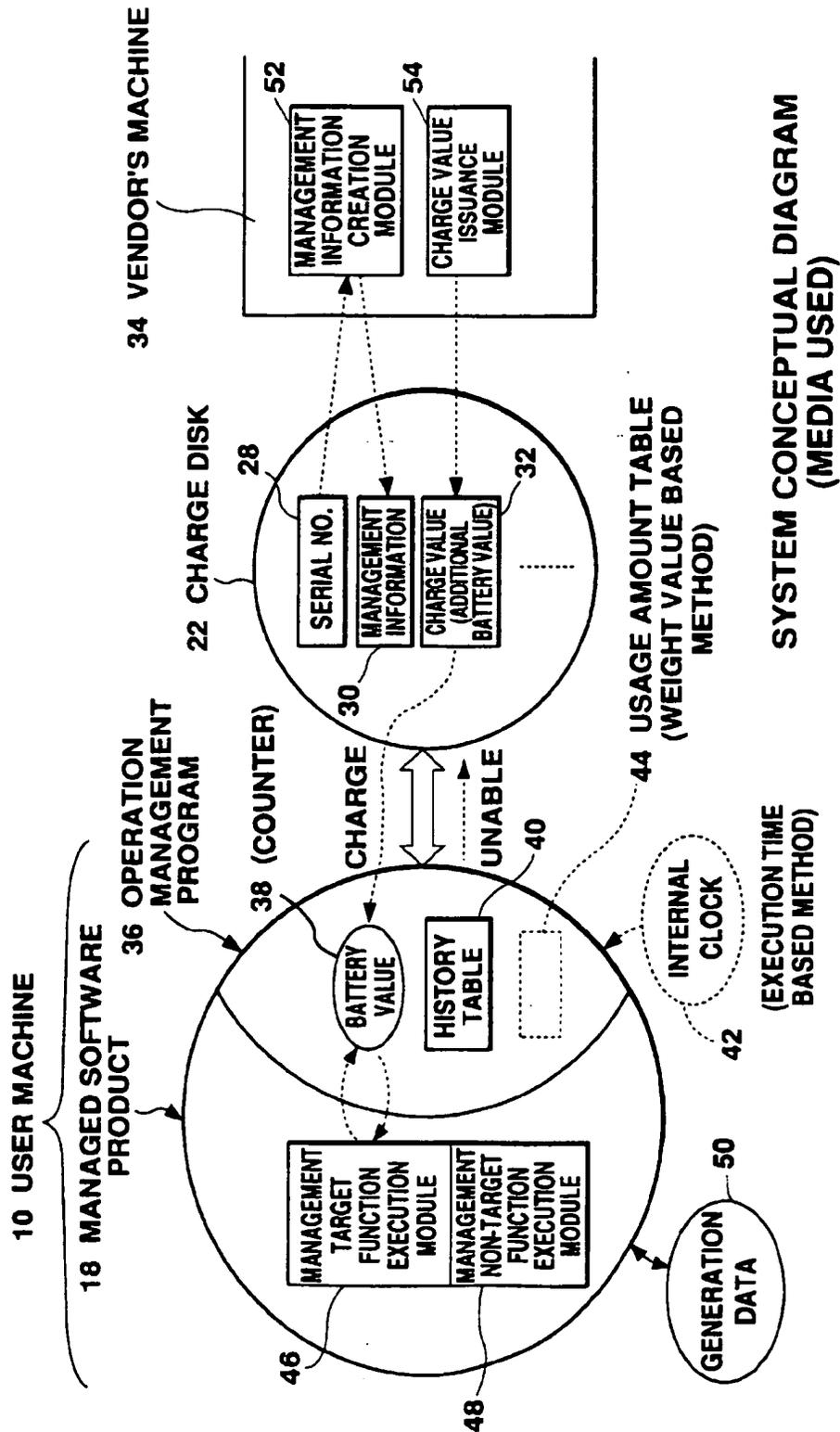


Fig. 3

44 USAGE AMOUNT TABLE

FUNCTION NAME	USAGE AMOUNT (WEIGHT VALUE)
.....

Fig. 4

40 HISTORY TABLE

FD SERIAL NO.	CHARGE DATE/TIME	CHARGED VALUE
.....

Fig. 5

MANAGEMENT TARGET FUNCTION EXECUTION
(EXECUTION TIME BASED METHOD)

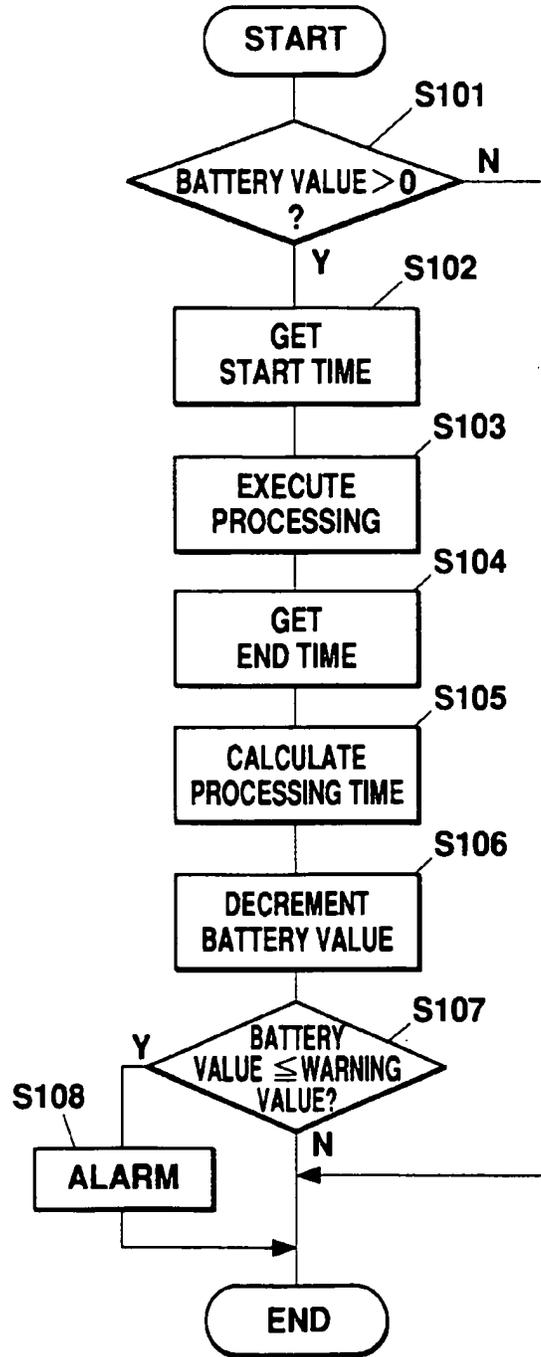


Fig. 6

MANAGEMENT TARGET FUNCTION EXECUTION (WEIGHT VALUE BASED METHOD)

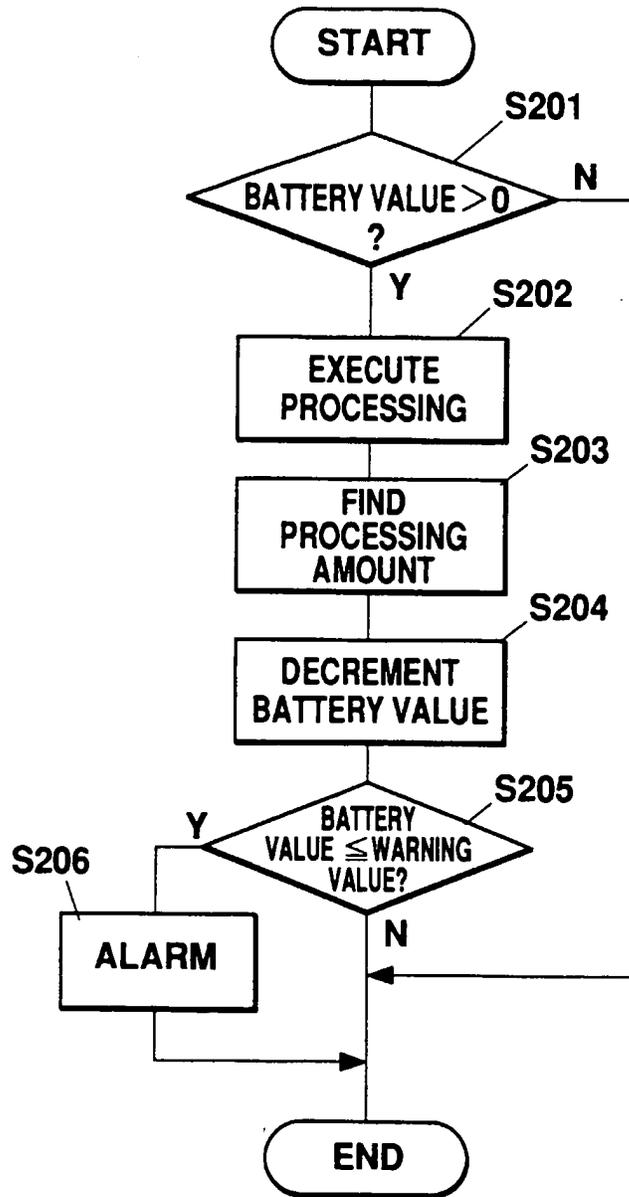


Fig. 7

CHARGE DISK READ PROCESSING

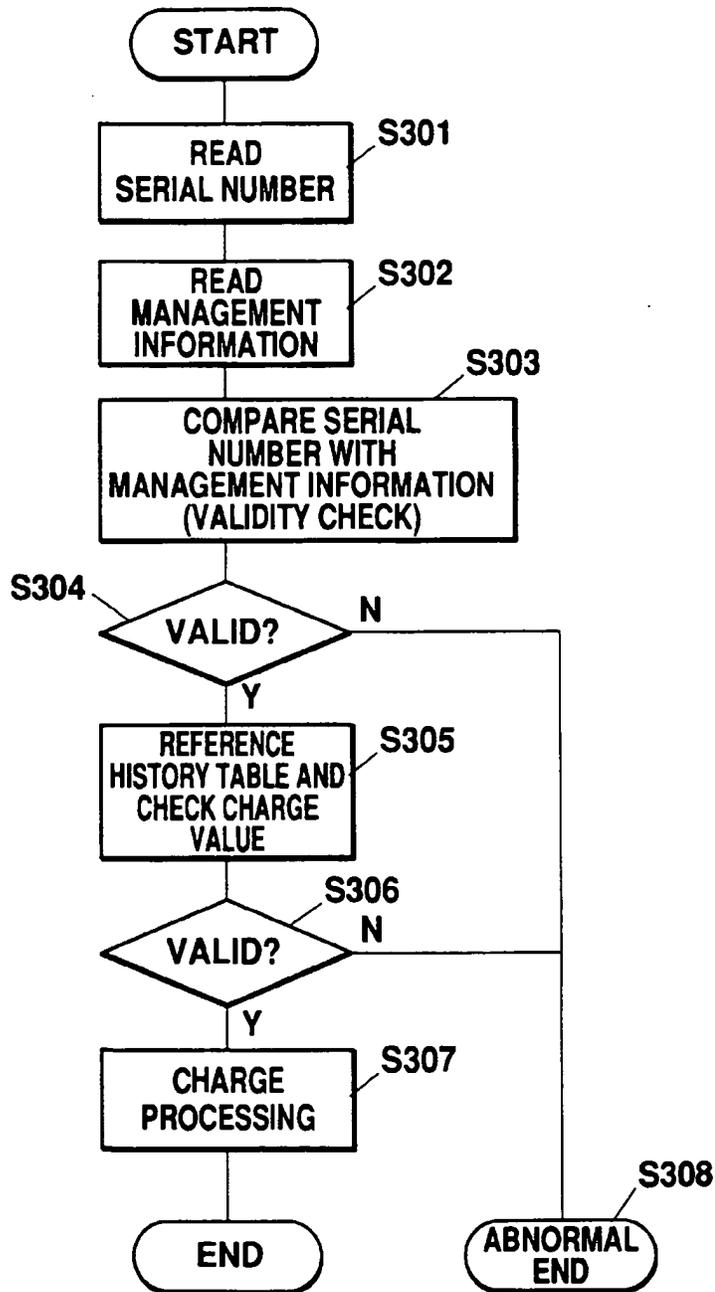


Fig. 8

CHARGE PROCESSING

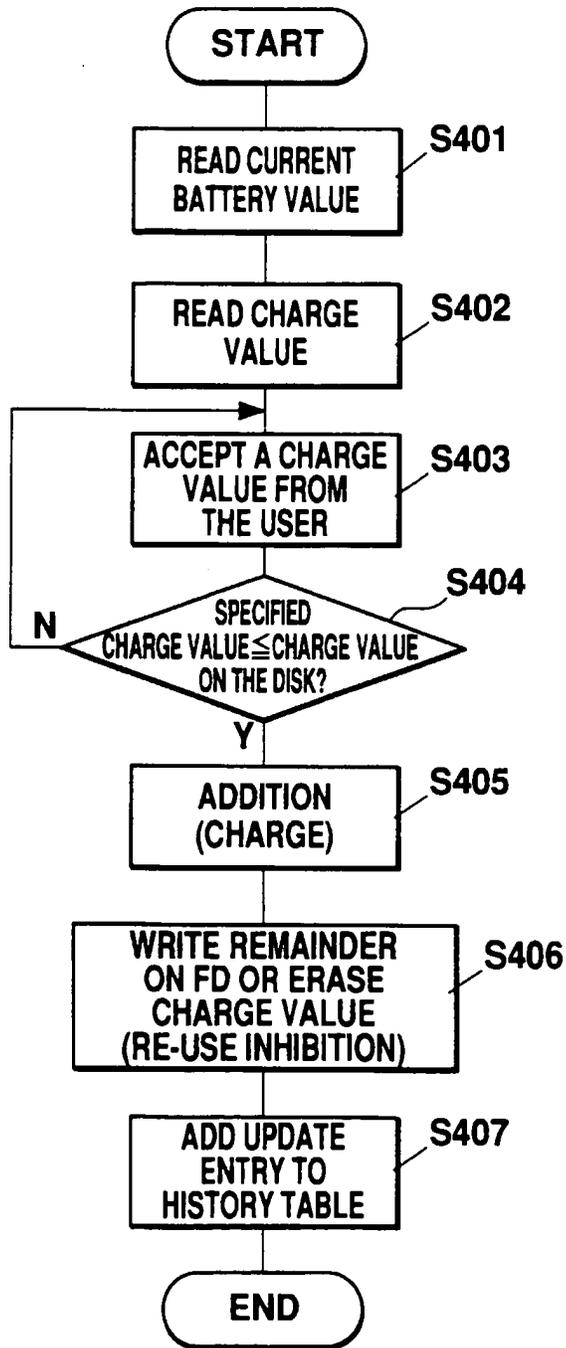


Fig. 9

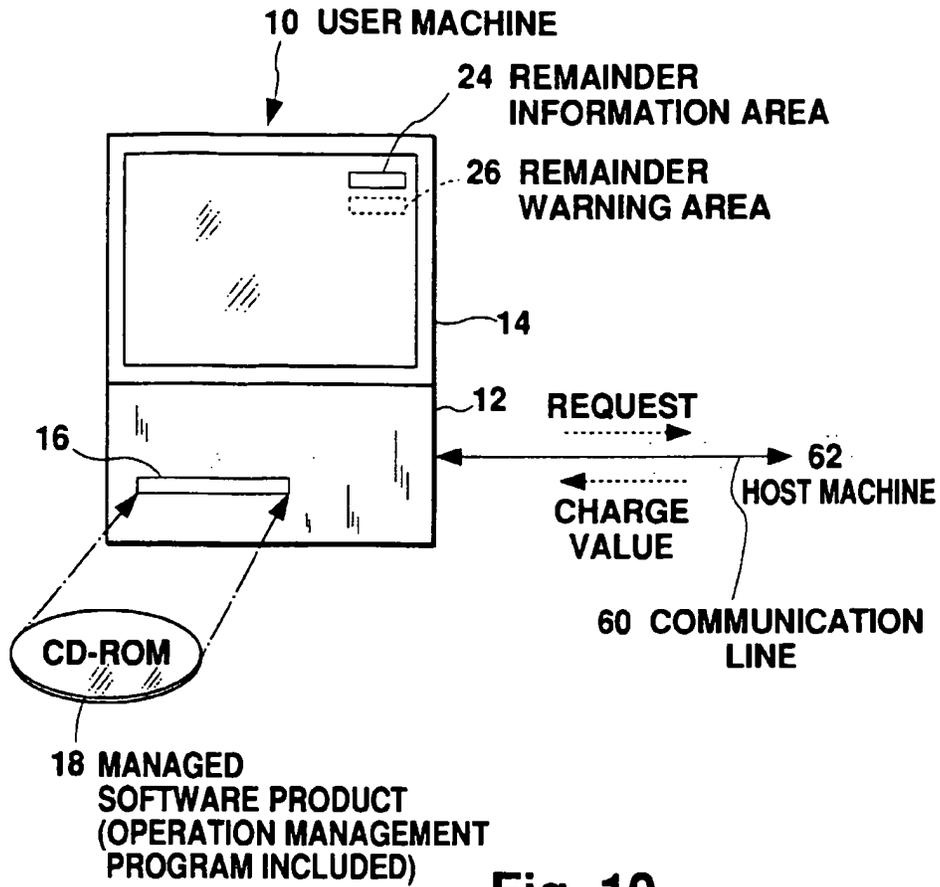


Fig. 10

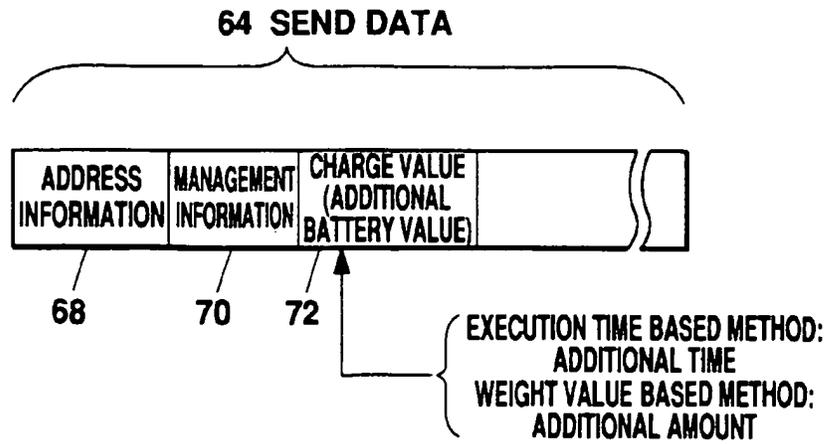
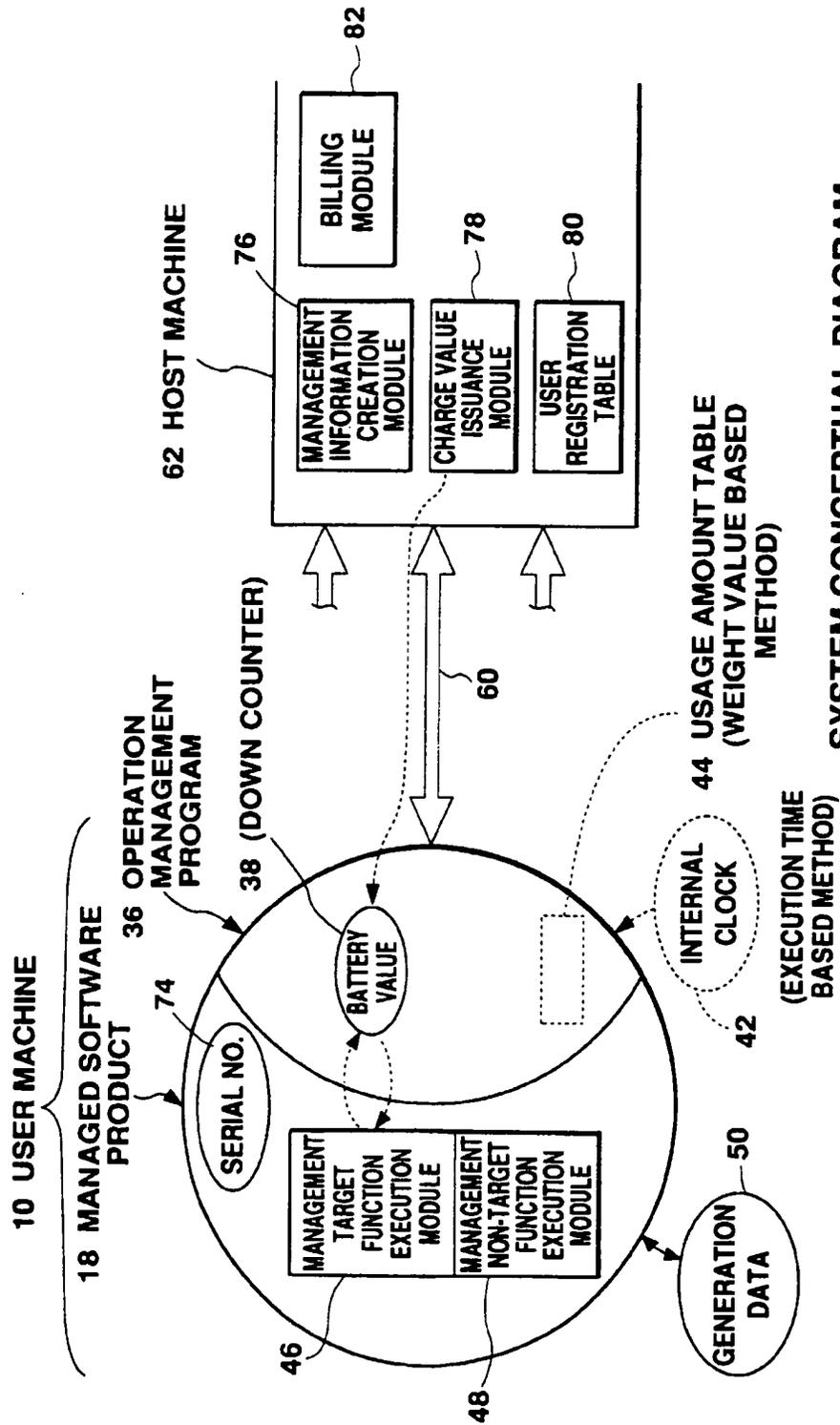


Fig. 11



SYSTEM CONCEPTUAL DIAGRAM (COMMUNICATION USED)

Fig. 12

80 USER REGISTRATION TABLE

80A ID	80B USER NAME	80C REQUESTED CHARGE VALUE
.....

Fig. 13

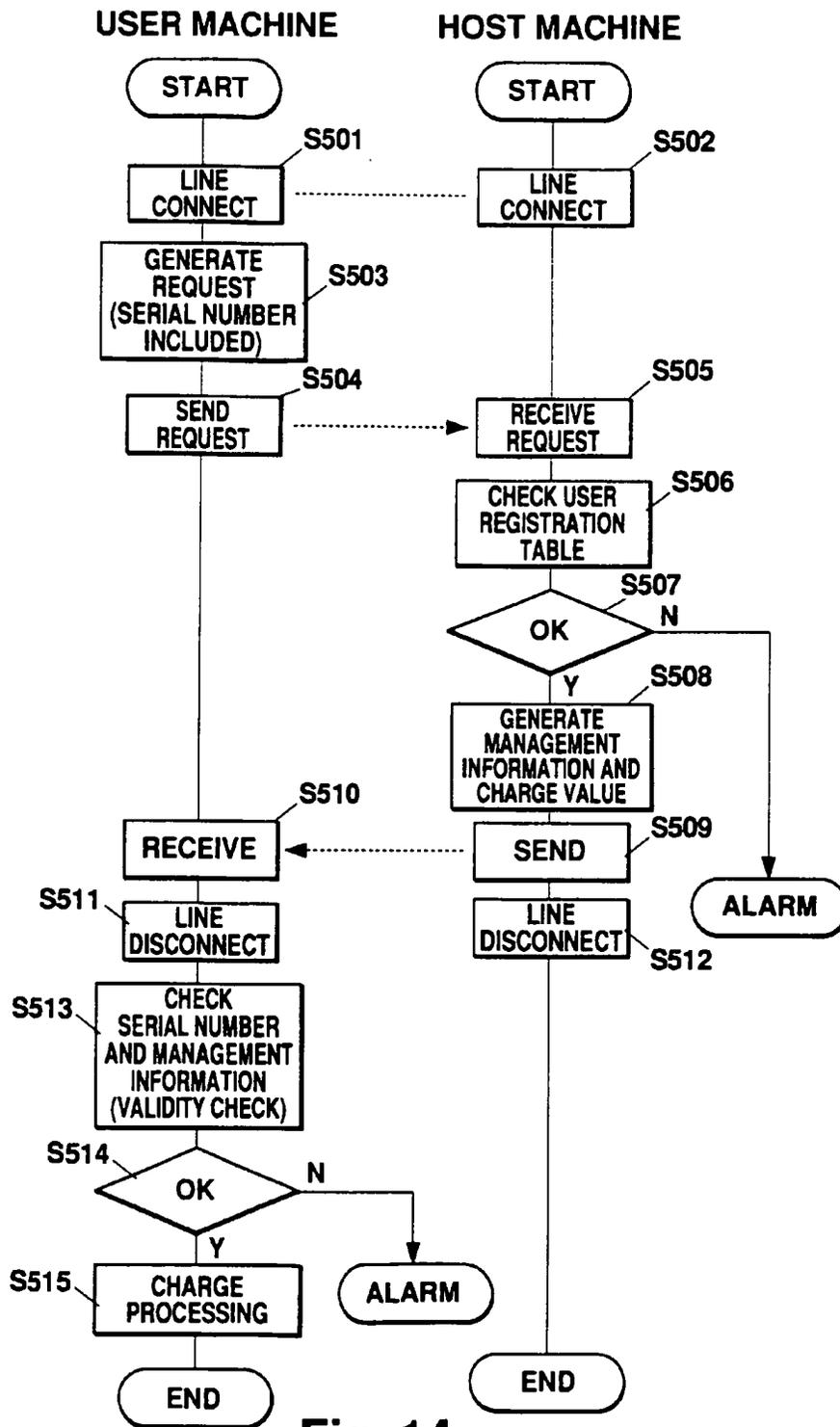


Fig. 14

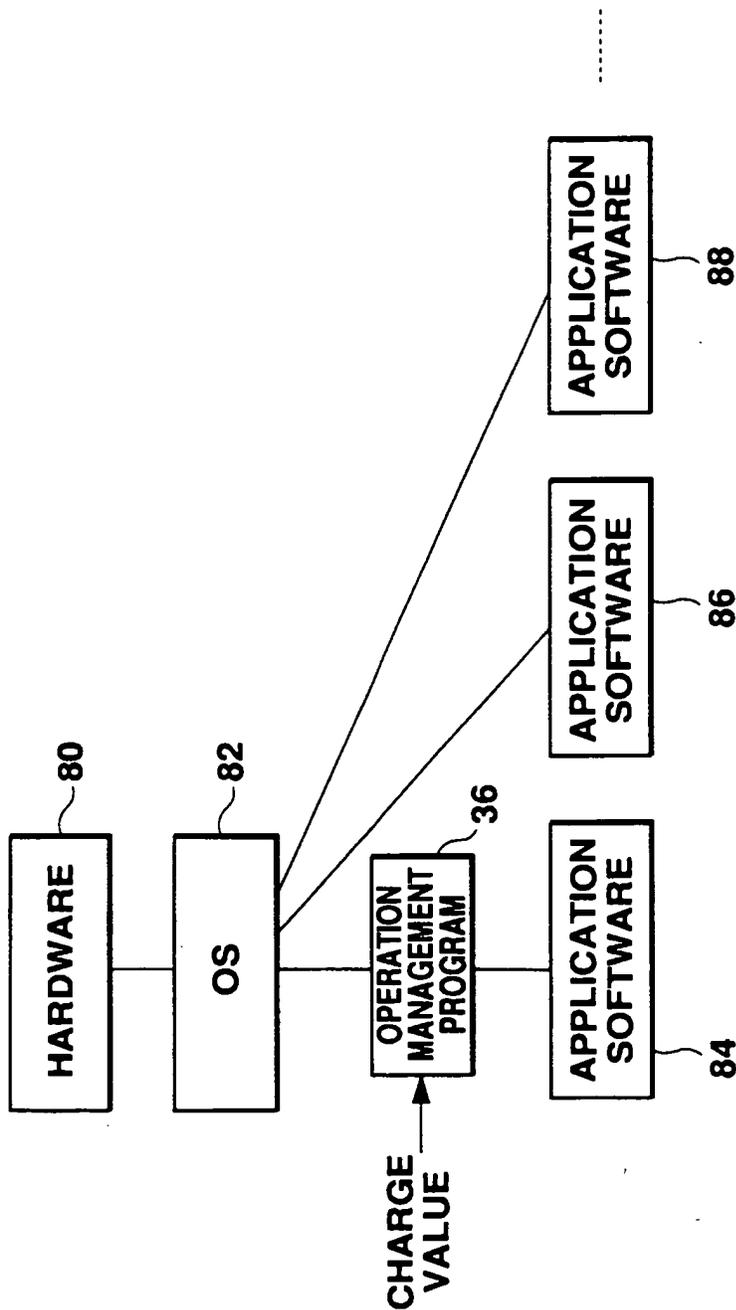


Fig. 15

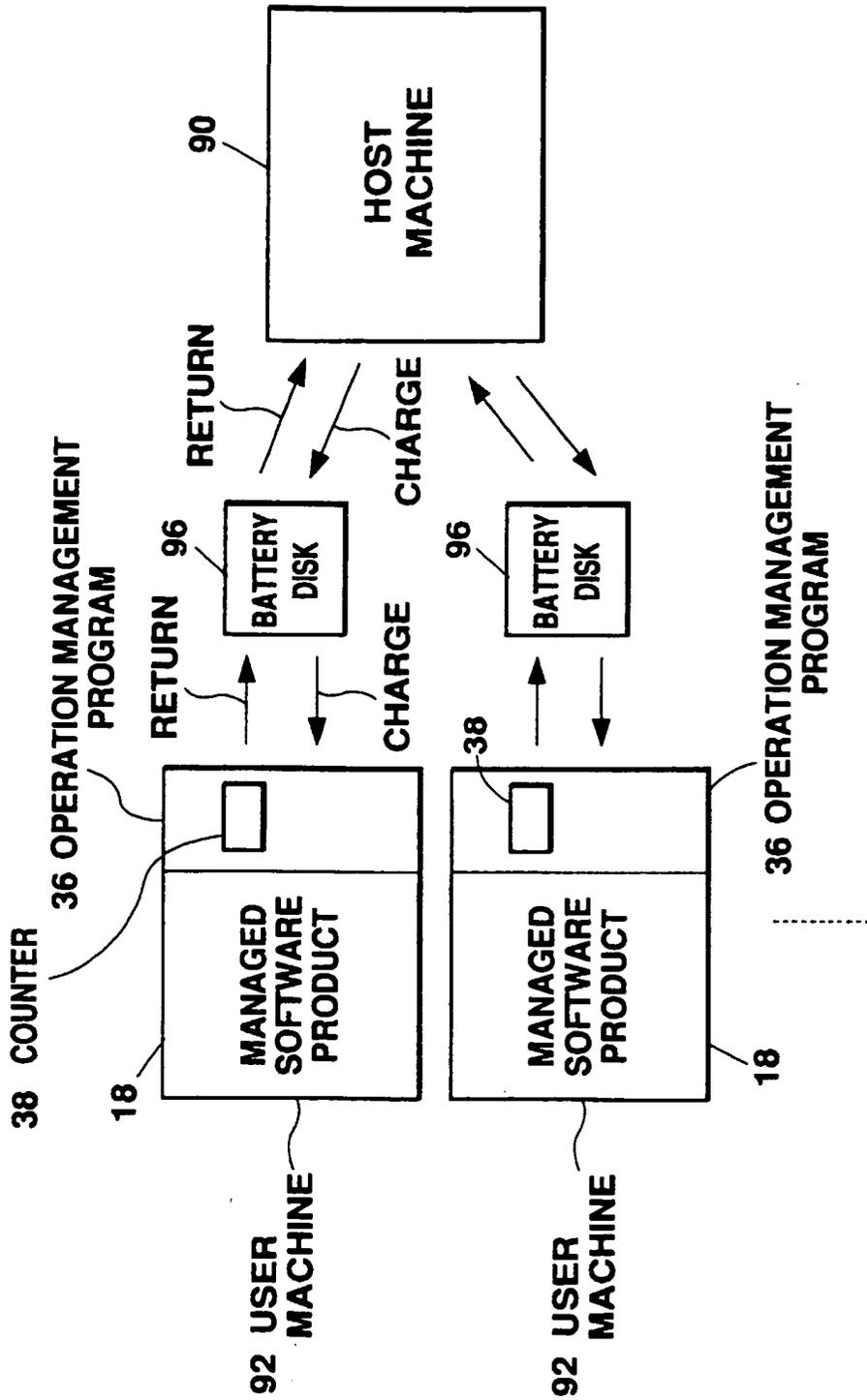


Fig. 16

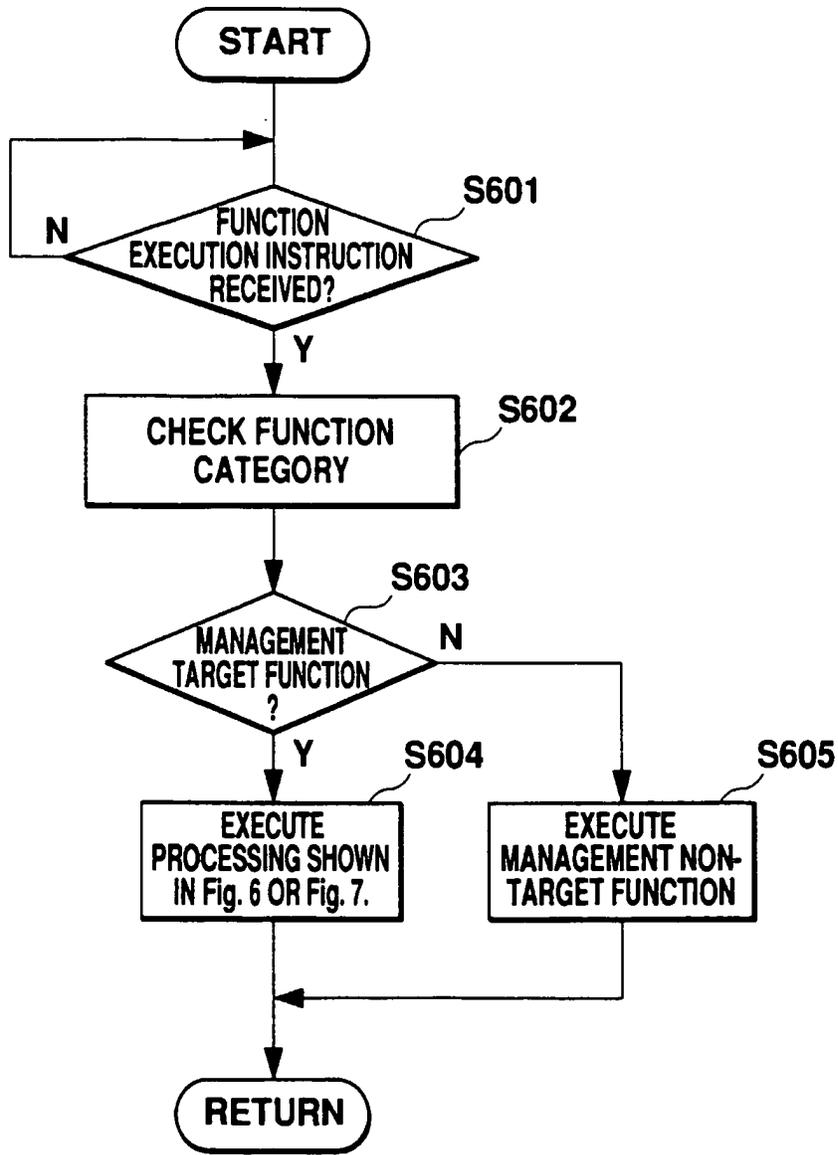


Fig. 17



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) EP 0 840 194 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
06.05.1998 Bulletin 1998/19

(51) Int. Cl.⁶: G06F 1/00

(21) Application number: 97108754.9

(22) Date of filing: 02.06.1997

(84) Designated Contracting States:
DE FR GB
Designated Extension States:
AL LT LV RO SI

(72) Inventors:
• Uranaka, Sachiko
Tokyo (JP)
• Kiyono, Masaki
Kamakura-shi, Kanagawa-ken (JP)

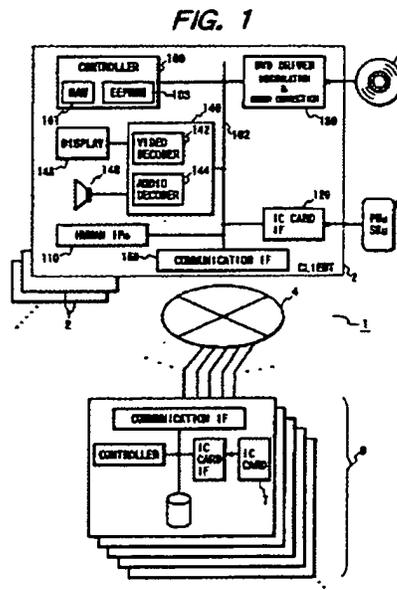
(30) Priority: 29.10.1996 JP 286345/96

(71) Applicant:
MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.
Kadoma-shi Osaka (JP)

(74) Representative:
Pellmann, Hans-Bernd, Dipl.-Ing. et al
Patentanwältsbüro
Tiedtke-Bühling-Kinne & Partner
Bavariastr. 4
80336 München (DE)

(54) System and method for controlling the use of a package of distributed application software

(57) A system for permitting only an authentic user to play a desired application contained in a distributed application package in one of predetermined operation, e.g., free play mode, charged mode, limit-attached play mode, etc. The system comprises a client for playing an application under the control of a server connected with the client through a communication network. The application package (the volume) includes a distribution descriptor which contains mode codes assigned to the volume and the applications of the volume. The data of distribution descriptor is decided and stored in the descriptor at the time of distribution of the volume. This feature makes the system flexible. There is also disclosed a system operable without communicating with a server.



EP 0 840 194 A2

Description**BACKGROUND OF THE INVENTION**

5 1. Field of the Invention

The invention generally relates to a security system and, more specifically, to a method and system for permitting an authentic user to use charged information which has been distributed via package or transmission media while charging and controlling the use of distributed charged information.

10

2. Description of the Prior Art

In order to use charged information such as music, movies, games, etc. provided by information providers that provide various programs of such charged information, a user has generally to take two steps. In the first step (or obtaining step), the user obtains a desired program from one of the information providers by purchasing a package media such as an FD (floppy disc), an optical disc (e.g., CD-ROM (compact disc read only memory) and DVD (digital versatile disc or video disc)), etc. on which the desired program is recorded (off-line distribution or obtaining) or by down loading the desired program from the server computer of an information provider through a predetermined procedure (on-line distribution or obtaining). In case of the on-line obtaining, the user may either play the program while obtaining it (i.e., the two steps are executed in parallel) or store the program while obtaining it in the first step and execute the program later as the second step (or using step). In case of the off-line obtaining, in the second step the user loads the obtained recording media into an appropriate device and directly plays (or executes) the program or once stores the program into the memory of the device and then plays the program.

Japanese Patent unexamined publication No. Hei7-295674 (1995) discloses a security system for use in the second or using step for a CD-ROM. In this system, the user can use encrypted information which is recorded together with a public key of a toll center (a center public key) on a CD-ROM by encrypting with the center public key and sending a code of desired program included in the information and a user-generated key to the information provider and by decrypting the information with an encryption key which has been encrypted with the user-generated key and sent by the information provider. However, the identity of the user is not verified, permitting a mala fide user who have obtained other person's CD-ROM to use it. Further, the center public key is pressed together with the encrypted information on the CD-ROM. This makes it difficult to change the center public key. Also, this causes different providers who probably want to use different center public keys to force the CD-ROM manufacturer to use different masters (or stampers) in pressing the CD-ROMs.

Japanese Patent unexamined publication No. Hei7-288519 (1995) discloses a security system for use in both the first and second steps. However, this system is only applicable to a system in which charged information is distributed on line.

Japanese Patent unexamined publication No. Hei8-54951 (1996) discloses a system in which the quantity of used software is monitored, and further software use by the user is impeded if the quantity exceeds a predetermined quantity. Since a dedicated hardware is necessary for impeding of software use, this system is only suitable for the use in a server in a on-line distribution system.

There is also a system for permitting a user to use, only for a trial period, software which has been distributed with data defining the trial period. In this system, a mala fide user may make the software reusable by installing the software again or setting the user system clock for a past time.

There are these and other programs in the art. It is an object of the invention to provide a system for permitting only an authentic user (a user who have legally obtained charged information either on line or off line from an information provider) to use the charged information without any limitation, charging for each time of its use, or within the tolerance of a use-limiting factor (e.g., the quantity used, the days elapsed since the day of its purchase or the current date) according to the type of the charged information.

50 **SUMMARY OF THE INVENTION**

According to the principles of the invention, it is assumed that charged information or an application package is distributed, either via package (or recording) media or via transmission media, together with at least control information such as a media title and a media code, etc. However, an illustrative embodiment will be described mainly in conjunction with charged information recorded on and distributed by means of the DVD.

For any type of charged information, charged information has been encrypted with a key and recorded on a DVD when obtained by a user. If distributed charged information to be played is of the limitlessly playable type, the charged information processing is achieved in the following way: the key is first obtained in a user public key-encrypted form from

the DVD on which the key has been recorded at the time of selling the DVD; the user public key-encrypted key is decrypted with a user secret key stored in a IC card into a decrypted key; and the encrypted charged information is decrypted with the decrypted key and consumed (that is, played or executed). The user-public key-encrypted key may be obtained on line from the server serving the client (device).

5 If distributed charged information to be played is of the usage-sensitive charging type, the user is charged for each time of using the information. In this case, prior to processing the charged information, the client double-encrypts and sends a user's credit card number to one of the to 11 servers of the provider of the information; the server adds an amount (e.g., play time or duration) used associated with the information to the value in a total amount (software meter) field in a volume data table, and sends the updated total amount value to the client; and the client displays the updated
10 total amount. Then the client starts the charged information processing.

If distributed charged information to be played is of the limit-attached type, that is, the use of the information is to be limited by the tolerance of a certain limiting factor concerning the Information consumption, then the client is permitted to consume the charged information only if the use-limiting factor is within the preset limit. In case of this type of charged information, prior to processing the charged information, the client sends the identifier (ID) code of a user specified application which is recorded on the DVD to the server; on receiving the ID code the server tests if the use-limiting factor associated with the user specified application is within the preset limit; if not, then the server informs the client of the test result, and the client displays the test result; if the test was successful, then the server updates the meter (or integrated value) of the use-limiting factor and sends the updated value to the client; and in response to the reception of the updated value the client displays the updated value. Then the client starts the charged information processing.
15
20

BRIEF DESCRIPTION OF THE DRAWING

Further objects and advantages of the present invention will be apparent from the following description of the preferred embodiments of the invention as illustrated in the accompanying drawings. In the drawing,
25

FIG. 1 is a block diagram showing an arrangement of a system for permitting a user to use a distributed application package on the terms of use of the package with a higher security according to a first illustrative embodiment of the invention;

FIG. 2 is a diagram showing an exemplary structure of an application (or a charged information) package recorded on a DVD used in the inventive system;

FIGs. 3 and 4 are diagrams showing, in a detailed form, exemplary data structures of the volume descriptor 22 and the distribution descriptor 23, respectively;

FIG. 5 is a flow chart of a volume control program for playing the application(s) recorded on the DVD according to the principle of the invention;

FIG. 6A is a diagram showing an exemplary structure of a volume data table stored in a server shown in FIG. 1;

FIG. 6B is a diagram showing an exemplary structure of a application data table stored in a server 8;

FIG. 7 is a diagram showing a structure of a server table 75 stored in the EEPROM 103 of the client 2;

FIGs. 8A and 8B are flow charts of initial routines executed interactively by the client 2 and the server 8, respectively, at the beginning of the processes 650, 700 and 800.

FIG. 9 is a flow chart showing a procedure of a free play process shown as step 650 in FIG. 5, wherein connecting adjacent blocks by two flow lines indicates that each block is executed interactively by a client and an associated server;

FIGs. 10A and 10B are flow charts jointly showing a procedure formed of exemplary expected play time informing routines interactively executed;

FIGs. 11A and 11B are flow charts jointly showing a procedure formed of exemplary timed play and metered usage report routines interactively executed for playing an application while timing the duration and displaying a timed play duration after the play;

FIGs. 12A and 12B are flow charts jointly showing a procedure formed of exemplary timed application-play subroutines interactively executed for playing the application while timing the duration;

FIGs. 13A and 13B are flow charts jointly showing a procedure formed of alternative timed application-play subroutines interactively executed in which timing of play time is achieved with a timer in the client;

FIG. 14 is a flow chart of an exemplary application play subroutine called in steps 612 and 622 of FIGs. 12A and 13A, respectively, and executed by the controller 100;

FIG. 15 is a flow chart showing a procedure of a charged play process 700 shown as step 700 in FIG. 5.

FIGs. 16A and 16B are flow charts jointly showing a procedure formed of exemplary expected charge informing routines interactively executed;

FIGs. 17A and 17B are flow charts jointly showing a procedure formed of routines interactively executed in block 650 of FIG. 15;

FIGs. 18A and 18B are flow charts jointly showing a procedure formed of exemplary timed play and metered charge report routines interactively executed for playing an application while timing the duration and displaying a charge and a total amount of charges after the play;

FIG. 19 is a flow chart showing a procedure interactively executed by the client 2 and the server 8 in the operation block 800 of FIG. 5, wherein blocks connected with two flow lines indicates that operation of the blocks is done by the two elements 2 and 8;

FIGs. 20A and 20B are a key-encrypting key table and a user's public key table, respectively, stored in the server; and

FIG. 20C is a flow chart of a process for obtaining the application encrypting key K_v from the server 8;

FIG. 21 is a block diagram of an exemplary decipherer-built-in IC card IF according to the invention;

FIG. 22 is a diagram showing a K_v decoder used in place of the K_v decoder 126 of FIG. 21 in a system 1 using the cryptosystem of FIG. 20C;

FIG. 23 is a diagram for explaining the meanings of the terms-of-use (TOU) codes and the corresponding limit values;

FIG. 24 is a block diagram showing an arrangement of a system for playing a distributed application package on the terms of use of the package without communicating with any server according to a second illustrative embodiment of the invention;

FIG. 25 is a flow chart schematically showing an exemplary control program executed by the controller 100a shown in FIG. 24;

FIGs. 26 and 27 are flow charts showing an operation of a free play mode shown in step 650a of FIG. 25 in a detailed form and a further detailed form, respectively; and

FIG. 28 is a flow chart showing an operation of a limit-attached play mode shown in step 800a of FIG. 25.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

For the sake of better understanding of the following description, it will be useful to define some terms to be used.

Charged information provided by an information provider may be distributed off-line (in off-line distribution) or on-line (in on-line distribution). In off-line distribution, the charged information is recorded on package media or recording media, and distributed through the sales network of the provider, that is, sold at stores in the sales network. The package media include all sorts of portable recording media such as various types of magnetic discs, a variety of optical memory discs (e.g., CD, CD-ROM, DVD), and magnetic tapes and cartridges. In on-line distribution, the charged information is transmitted via transmission media from the servers at the service points of the provider and the distributors aligned with the provider to the client device (e.g., PC (personal computer)) of the user who requested the charged information, and stored in a recording media of the client (device). The transmission media include any telecommunication channels which permit data communication between the servers and the client device. The package media and the transmission media are hereinafter referred to en bloc as "distribution media".

The charged information may be any type of software such as music, movies, games, etc. which are each referred to as an "application" without discrimination. The distribution unit of charged information is referred to as a "charged information package" or an "application package". There may be included one or more applications in an application package.

The present invention relates to a system for permitting a user to use a distributed application package on the terms of use of the package with a higher security.

Embodiment I

For the purpose of simplicity, a first illustrative embodiment will be described in which package media, among other things, DVDs are used as distribution media.

FIG. 1 is a block diagram showing an arrangement of a system for permitting a user to use the application(s) recorded on a DVD on the terms of use of the DVD with a higher security according to the first illustrative embodiment of the invention. In FIG. 1, the system 1 comprises a client or DVD player 2 which plays a DVD 3, a telecommunication network 4, and a server 8 at a toll center of the provider 6 which provides the application package of the DVD 3.

FIG. 2 is a diagram showing an exemplary structure of an application (or a charged information) package 20 recorded on the DVD 3 used in the inventive system 1. In FIG. 2, the application package 20 comprises at least one application 21, a volume (or package) descriptor 22 comprising data concerning the application package 20, and a distribution descriptor 23 comprising data which is determined mainly at the time of, e.g., distribution or sales after the pressing of the DVD 3. (The volume descriptor 22 and the distribution descriptor 23 constitutes the volume control data of the volume 20.) In this embodiment it is assumed that a volume (or package) control program which controls the use of the application package 20 in cooperation with the server 8 is included in and distributed with the application package

20. Thus, the application package 20 further comprises the package control program 24 suited for the terms of use of the package 20. The application(s) 21, the volume descriptor 22 and the package (or volume) control program 24 are recorded in the data area of the DVD 3 at the time of manufacturing the DVD 3, while the distribution descriptor 23 is recorded in the burst cutting area at the time of, e.g., sales of the DVD 3.

5 FIGs. 3 and 4 are diagrams showing, in a detailed form, exemplary data structures of the volume descriptor 22 and the distribution descriptor 23, respectively. In FIG. 3, the volume descriptor 22 at least contains a volume identifier (VID_v) 25 which the title of the application package 20 is probably used for and which is the same as the application identifier if the package or volume 20 contains only one application; a provider identifier 26; volume creation date and time 27 which may be used for the base point by which volume expiration data and time as described later is determined; and volume effective date and time 28 indicative of date and time until which the volume 20 is available. If the volume 20 contains more than one applications, the volume descriptor 22 further contains application identifiers (AID_a's) 29.

10 In FIG. 4, the distribution descriptor 23 comprises the fields of: a volume issue number (NO_{v,i}) 30 which contains a serial number given to each of the distributed application packages of an identical volume identifier (volume ID or title) VID_v in the order of distribution; a server public key (PK_s) 31 the data of which is given by the server 6 at a toll center of the provider 6; a PK_u (user-public-key)-encrypted application-encrypting key (K_v) 32; and sales date and time 33. The key PK_s 31 field contains a key which has been used in encrypting each application 21 in the package 20 and which has been encrypted with a user public key (PK_u) of the user who has legally obtained the package 20. Appropriate data are recorded in all of the fields 30 through 34 at the time of distribution of the package 20, i.e., at the time of sales of the DVD 3 in this embodiment.

15 The distribution descriptor 23 further comprises the field 34 of terms-of-use code (mode code) plus limit value for the volume (the volume limit value field) and, for each of the application IDs 29, the fields 35 of terms-of-use code plus limit value for the application ID 29 (application limit value field). If terms of use are set only to the volume 20, there is no need of the field 35. If terms of use are set to each application, the field is empty.

25 FIG. 23 is a diagram for explaining the meanings of the terms-of-use (TOU) codes and the corresponding limit values. In FIG. 23, the terms-of-use code may be, e.g., one byte in length. The higher digit (X) of the TOU code indicates the target to which the terms of use is applied as shown in table 36. That is, higher digits of 0, 1, 2,... indicate that the TOU codes beginning with those digits are for the entire volume, application 1, application 2 and so on. The lower digit (Y) of the above mentioned terms-of-use code indicates the terms of use of the package 20 or the application 21 to which the code is set, and is directly followed by a corresponding limit value as shown in table 37 of FIG. 23. Specifically, the terms-of-use code (or TOU code) of 00H means, for example, that the volume 20 is usable freely after distribution. The value '31H' means, for example, that the application 3 to which the TOU code is set can be used by paying per unit of play duration. The lower digit of 2H or more means that the volume 20 or the application to which the TOU code is set can be used freely until the corresponding limit value are reached, which disables further use. As seen from the table, the use-limiting factors determined by the TOU codes whose lower digits are 2H to 5H are the current date and time, the expiration date and time, the amount of used period, and the access count, respectively.

30 Since the data of the distribution descriptor 23 can be set as described above, this provides both the providers and the users with more flexibility than conventional system can provide.

40 Again in FIG. 1, the DVD player 2 comprises a controller 100 for controlling the entire DVD player 2; data bus 102 connected with the not-shown CPU (central processing unit), not-shown ROM (read-only memory), RAM (random access memory) 101, and EEPROM (electrically erasable programmable ROM) 103 included in the controller 100; human interfaces (IFs) 110 including input devices such as a keyboard, a voice recognition device, a mouse, a remote controller, etc.; an IC card interface (IF) 120 for connecting the bus 102 with the ROM (not shown) in a IC card 5; a DVD driver 130 for reading out the data recorded on the DVD 3 and for demodulating and error-correcting the read data; a video and audio output IF 140 for receiving a MPEG 2 bit stream and outputting a video and audio output signals; a display device 146; a loudspeaker 148, and a communication IF 150 for communicating through the public telecommunication network 4. The IC card 5 stores a user's password PW_u and a user's secret key SK_u which corresponds to the user's public key PK_u mentioned in conjunction with the PK_u-encrypted AP-encrypting key (K_v) contained in the field 32 of the distribution descriptor 23 recorded in the burst cutting area of the DVD 3. The video and audio output IF 140 includes a MPEG 2 video decoder 142 and a MPEG 2 audio decoder 144.

45 As for obtaining the DVD 3, there may be some ways. If one is to buy a DVD 3, e.g., at some book store or through mail order, he or she has to have the PK_u-encrypted version of an application-encrypting key (K_v) recorded in the burst cutting area of the desired DVD 3 by notifying his or her public key PK_u which corresponds to his or her secret key SK_u stored in the IC card 5. If one is a member of a DVD distribution service, he or she can obtain a DVD with a PK_u-encrypted AP-encrypting key recorded without notifying the PK_u each time of obtaining because he or she must have notified the PK_u when he or she applied for the service.

50 In operation, the user first sets a desired DVD 3 in the DVD driver 130 of the DVD player 2, and issues a start command to the DVD player 2 through an appropriate human IF 110. In response to a receipt of the start command, the

controller 100 reads the volume control program 24 from the data area of the DVD 3 through the DVD driver 130 while loading the read program 24 into the RAM 101 of the controller 100, and then executes the volume control program 24.

FIG. 5 is a flow chart of the volume control program 24 for playing the application(s) 21 recorded on the DVD 3 according to the principle of the invention. In FIG. 5, the controller 100 first checks the AID1 field to see if the volume 20 contains a single application in step 500. If not, then the controller 100 displays the application IDs in the field 29 and prompts the user to select a desired one of the applications in step 502, and waits for the selection in step 504. If any application is selected in step 504, the controller 100 registers the application ID of the application as the application to be played in step 506 and proceeds to step 508 to check the field 35 of the terms-of-use (TOU) code plus limit value for the selected application to see if the field is empty. If so, the controller 100 proceeds to step 510 to read the volume limit field 34.

On the other hand, if the test result is YES in step 500, then the controller 100 registers the volume ID as the application to be played in step 512, and reads the volume limit value 34 in step 510.

If the step 510 is completed or the test result of step 508 is NO, then the controller 100 checks the terms-of-use (TOU) code to see if the lower digit of the TOU code is 0 in step 514. If so, then the controller 100 plays an application free of charge in step 650, and otherwise makes another check to see if the lower digit of the TOU code is 1 in step 516. If so, the controller 100 plays an application in a usage-sensitive charging in step 700, and otherwise (if the lower digit of the TOU code is 2 or more) play an application only when the software meter of a use-limiting factor is under a preset value in step 800. On completing any of the steps or processes 650 through 800, the controller 100 ends the program 24. Thus, the DVD player 2 plays a program specified by the user according to the terms of use determined by the TOU code which has been set to either the application package or the specified application.

The processes 650, 700 and 800 are executed interactively with an associated server 8. The servers 8 need various data for executing these processes, and store such data in the form of tables.

FIG. 6A is a diagram showing an exemplary structure of a volume data table stored in a server 8. In FIG. 6A, Each of the records of the volume data table 60 comprises volume ID (VID_v) and issue No. (NO_{v-i}) fields. The combination of VID_v and NO_{v-i} serves as the user ID of the user of the application package 20 or the DVD 3. For this reason, the table 60 has, for the members or subscribers of DVD distribution service or the like, personal data fields which contains, for example, a member ID, a name, an address, etc. Each record further comprises a volume minute meter field (VM-METER_{v-i}) containing a software meter of play duration in minute which is attached to (or associated with) the volume 20; a volume charge meter (VC-METER_{v-i}) containing a software charge meter which is attached to the volume 20; a limit value (LV_{v-i}) containing a limit value associated with the TOU code (e.g., the effective date and time, the allowable expiration date and time, the allowable access, etc.); a limit value meter (LV-METER_{v-i}); an application ID (AID_{v+i-a}) field containing the title of the application; an application minute meter (AM-METER_{v+i-a}) field containing a software meter of play duration in minute which is attached to the application of AID_{v+i-a}; an application charge meter (AC-METER_{v+i-a}) field for a software meter of play duration in minute which is attached to the application of AID_{v+i-a}; a limit value (LV_{v+i-a}) containing a limit value associated with the TOU code; and a limit value meter (LV-METER_{v+i-a}).

FIG. 6B is a diagram showing an exemplary structure of an application data table stored in a server 8. In FIG. 6B, the application data table 70 comprises the fields of, for example, an application code (ACODE_n), an application title (AID_n), a duration (D), a rate-per-access (RATE/ACCESS), an access count, a minute meter, etc. The duration is a period of time what it takes to play the application. The rate per access is a charge for a play of the whole application, which is used for informing the user of an expected play duration prior to a play. The rate per unit time is a charge for a unit time of play, which is used for the calculation of a charge for an actually timed play duration. The access count and minute meter fields contains the number of accesses to the application and a total amount of play time, which are not necessary for the present invention but will be used in statistical calculations for the analysis of, e.g., the tastes.

FIG. 7 is a diagram showing a structure of a server table 75 stored in the EEPROM 103 of the client 2. In FIG. 7, the fields of the table 75 comprises a server public key (PK_s), a server ID (SID_s), a server network address (SADD_s), etc. this table 75 is used for associating the sever public key (PK_s) contained in the distribution descriptor 23 recorded in the burst cutting area of the DVD with the ID and the network address.

Play an Application Free of Charge

The initial routines of the processes 650, 700 and 800 are the same.

FIGs. 8A and 8B are flow charts of initial routines 80a and 80b which are executed interactively by the client 2 and the server 8, respectively, at the beginning of the processes 650, 700 and 800. In FIG. 8, the controller 100 of the client or the DVD 2, in step 82, sends a service request with the network address CADD_c of the client or DVD 2, the TOU code plus limit value, the volume ID (VID_v), the issue number (NO_{v-i}), the application ID (AID_{v+i-a}), and other data to the associated server 8 the ID of which is SID_s (SID_s is obtained from the table 75 in FIG. 7 by using the public key recorded on the DVD 3), and in step 92 waits for a response from the server (SID_s) 8. If there is a response from the server (SID_s), the client 2 proceeds to the next step through a circle with "A" therein.

On the other hand, in FIG. 8B, the server 8 of SID_c receives the message from the client 2, that is, the service request and the accompanying data and stores data in a predetermined location for subsequent use in step 84. Then, the server 8 searches the table 60 for a record which contains VID_v and NO_{v-1} in the volume ID and issue No. fields thereof, respectively in step 86. If the search is unsuccessful, then the server 8 adds the record for VID_v and NO_{v-1} and fills relevant fields with AID_{v-t_a} and a limit value, if any, in the table 60 in step 88, and proceeds to step 90. Also, if the search in step 86 is successful, the server 9 proceeds to step 90, where the server 8 selects a routine to execute next according to the value of the TOU code and enters the selected routine through a circle with "B" therein. In this case, if the TOU code = $x0H$ (x: an arbitrary HEX number, the letter H in the last position indicates that the preceding number is in hexadecimal), then a routine for playing an application free of charge is selected. If the TOU code = $x1H$, then a routine for playing an application in usage-sensitive charging is selected. If the TOU code $\geq x2H$, then a routine is selected which plays an application only if the software meter of a use-limiting factor is under a preset value.

FIG. 9 is a flow chart showing a procedure of a free play process shown as step 650 in FIG. 5, wherein connecting adjacent blocks by two flow lines indicates that each block is executed interactively by a client of $CADD_c$ and an associated server SID_c , as shown in detail later. If the TOU code is 0 in step 514 of FIG. 5, then the server ($CADD_c$) enters the free play process 650 as shown in FIG. 9, and the client and the server (SID_c) execute the initial routine 80 in block 660. In block 670, they execute an expected play time informing routine, that is, displays an expected play time before playing an specified application. In block 680, they execute an application play and metered play time report routine. Since the routine 80 has been detailed in FIG. 8, the expected play time informing routine and the application play and metered play time report routine will be detailed in the following.

FIGs. 10A and 10B are flow charts jointly showing a procedure formed of exemplary expected play time informing routines 97a and 97b interactively executed by the client 2 and the associated server 8, respectively. In FIG. 10B, the server 8 retrieves the duration (D_n) of the application of AID_{v-t_a} from the table 70 in a well known manner in step 91. In the next step 92, the server 8 calculates an expected total amount of play time according to the value of the TOU code. Specifically, if the TOU code is $0xH$, then the client adds the duration (D_n) and the value of the VM-METER $_{v-1}$ field of the record identified by VID_v and NO_{v-1} in the table 60. If the TOU code is axH (a: the application number of the specified application in the volume), then the client adds the duration (D_n) and the value of the AM-METER $_{v-t_a}$ field of the record identified by VID_v , NO_{v-1} and AID_{v-t_a} in the table 60. Then the server 8 sends the result to the client whose network address is $CADD_c$ in step 93, and ends the process.

On the other hand in FIG. 10A, the client 2 receives the incoming message or the value of the updated meter in step 94. In the next step 95, the value is displayed as the total amount of usage. Then the client 2 ends the process.

In updating a relevant meter, a predetermined value of duration has been used in the just described routines of FIG. 10 (a preset value metering system). This arrangement is suited mainly for such applications as it takes a constant time to play, and will not cause a problem unless the user discontinues the play. From this point of view, it is preferable to actually measure the playing time in metering (a timed value metering system). However, it is also noted that the preset value metering system is useful in informing the user of expected play time prior to an actual playing.

FIGs. 11A and 11B are flow charts jointly showing a procedure formed of exemplary timed play and metered usage report routines 675a and 675b interactively executed by the client and the server, respectively, for playing an application while timing the duration and displaying a timed play duration after the play. In the routine 675, the client and the server call a timed application-play subroutine for playing the application while timing the duration (play time) in step 200.

Then the server 8 proceeds to step 210, where the client updates a relevant meter according to the TOU code in the same manner as in step 92 of FIG. 10B. Specifically, if the TOU code is $0xH$, then the play time is added to the value of the VM-METER $_{v-1}$ field of the record identified by VID_v and NO_{v-1} in the table 60. If the TOU code is axH (a: the application number of the specified application in the volume), then the play time is added to the value of the AM-METER $_{v-t_a}$ field of the record identified by VID_v , NO_{v-1} and AID_{v-t_a} in the table 60. Then the server 8 sends the play time and the value of the updated meter (i.e., the total amount of play time) to the client whose network address is $CADD_c$ in step 212, and ends the process.

On the other hand, the client 2, after step 200, make a test to see if there is a response from the server of SID_c in step 214. This step is repeated until the client 2 receives a call from the server 8, when the client 2 receives the incoming message or the value of the updated meter in step 216. In the next step 218, the client 2 displays the play time and the total amount of play time, and then ends the routine 675.

FIGs. 12A and 12B are flow charts jointly showing a procedure formed of exemplary timed application-play subroutines 205a and 205b executed by the client 2 and the server 8, respectively, for playing the application while timing the duration. The server 8 of SID_c waits for a notice in step 611 to see if the client has started playing the application. On the other hand, the client 2 of $CADD_c$ informs the server of a start of play in step 610 and immediately call an application play subroutine in step 612. This, causes the server 8 to start a timer in step 613, and waits for a notice of a stop of play from the client 2 in step 615. On completing the step 612, the client informs the server 8 of the stop of play in step 614. In response to this notice, the server 8 stops and reads the timer as the play time in step 617. After steps 614 and 617, the client and the server return.

Though the above described arrangement has used a timer of the server, it may be possible to use a timer of the client.

FIGs. 13A and 13B are flow charts jointly showing a procedure formed of alternative timed application-play subroutines 205ac and 205bc interactively executed by the client 2 and the server 8, respectively, in which timing of play time is achieved with a timer in the client. In the alternative subroutine 205a, the client 2 starts a timer in step 620, calls an application play routine in step 622, stops the timer in step 624, sends the play time to the server 8 in step 626, and then returns. On the other hand, the server 8, on entering the subroutine 295b, waits for a call from the client of CADD_c in step 621. If there is a call from the client 2, then the server 8 receives the play time in step 623 and then returns.

However, the arrangement of FIG. 13 has a possibility of permitting a mala fide user to manipulate the timer of the client 2. From this point of view, the arrangement shown in FIG. 12 is preferable to that of FIG. 13.

FIG. 14 is a flow chart of an exemplary application play subroutine called in steps 612 and 622 of FIGs. 12A and 13A, respectively, and executed by the controller 100.

Prior to the description of the flow chart, we define some notation concerning encryption and decryption. If encrypting X with a key EK according to an encrypting algorithm e yields Y, then it is expressed as:

$$e(EK, X) = Y.$$

Similarly, if decrypting Y with a key DK according to a decrypting algorithm d yields Z, then it is expressed as:

$$d(DK, Y) = Z.$$

Assuming that the algorithms e and d and the keys EK and DK correspond each other, that is, $d(DK, Y) = X$, it follows that

$$d(DK, e(EK, X)) = X.$$

Returning now to FIG. 14, the controller 100 read the PK_v-encrypted application-encrypting (AP-encrypting) key (K_v) or e1(PK_v, K_v) from the filed 32 of the distribution descriptor 23 of the DVD in step 602. Here,

$$v = 1, 2, \dots, V,$$

where V is the number of kinds of the application package. This indicates that different application-encrypting keys K1 through K_v is assigned to respective kinds of applications, that is, volume VID1 through VID_v.

In the next step 604, the user secret key SK_u is read from the IC card 5. In the next step 606, the PK_v-encrypted AP-encrypting key e1(PK_v, K_v) is decrypted with the user secret key SK_u to obtain the application encrypting key K_v. Then in the next step 608, the K_v-encrypted application (AP), i.e., e(K_v, AP) which is recorded on the DVD 3 is decrypted with the obtained AP-encrypting key K_v to obtain $d(K_v, e(K_v, AP)) = AP$, while passing the obtained application data to the video and audio output IF 140. The obtained application data has the form of an MPEG 2 bit stream. The video and audio output IF 140 converts the MPEG 2 bit stream of the application data into video and audio output signals through MPEG 2 video and audio decoding. The video and audio output signals are applied to the display device 146 and the loudspeaker 148, respectively.

Play an Application in Usage-sensitive Charging system

FIG. 15 is a flow chart showing a procedure of a charged play process 700 shown as step 700 in FIG. 5, wherein connecting adjacent blocks by two flow lines indicates that each block is executed interactively by a client of CADD_c and an associated server of SID_s. In FIG. 15, the client 2 enters the process 700 via step 516 of FIG. 5 and proceeds to block 630, where the client 2 and the associated server 8 execute the initial routine 80. In the next block 640, the client 2 displays an expected charge and a total amount of charges received from the server 8, and let the user decide whether to play the desired application.

FIGs. 16A and 16B are flow charts jointly showing a procedure formed of exemplary expected charge informing routines 640a and 640b interactively executed by the client 2 and the associated server 8, respectively. The routines 640a and 640b are very similar to the routine 97 except that in the routine 640, the DURATION (D_n) or "play time" has been replaced with RATE PER ACCESS and "charge"; between steps 92a and 93a, there has been added a step 641 of the server generating and storing a pseudo random number R in a memory location R'; in step 93a, the server sends the pseudo random number R as well; between steps 94 and 95a there has been added a step 643 of the client storing the received pseudo random number R in a memory location R" for subsequent use. The replacement of DURATION (D_n) with RATE PER ACCESS is achieved by accessing a RATE PER ACCESS field 74 instead of a DURATION field

73 in table 70. Further, in the routine 640 there have been added the following steps: in step 644 following the step 96a, the client 2 makes a check to see if the user decides to play the application; if not, the client 2 sends a quit message to the server of SADD_s in step 645, and ends the routine 640; on the other hand, in step 642 following the step 93a, the server 8 of SID_s waits for a call from the client 2 of CADD_c; on receiving a call from the client, the server makes another check in step 646 to see if what has been received is a quit message; if so, the client ends the routine 640; and if the user decided to play the application in step 644, which means that what the server has received is not a quit message but an encrypted credit card number as seen from the description below, then the client 2 and the server 8 proceed to the step 650 of FIG. 15.

In the next block 650, the server 8 obtains a user's credit card number (CCNOu) through the client 2 keeping the security of the card number as shown in FIGs. 17A and 17B. In step 647, the client 2 encrypts the credit card number of the user which has been input by the user through a human IF 110 with a key, i.e., the pseudo random number R which has been stored in a memory location R' in step 643 of FIG. 16A to obtain e2(R, CCNOu). In the next step 648, the client 2 further encrypts R + e2(R, CCNOu) with another key or a server public key read from the distribution descriptor 23 recorded in the burst cutting area of the DVD to obtain

$$e1(PK_s, R + e2(R, CCNOu)).$$

In the next step 649, the client 2 sends the encrypted data to the server 8. Through step 646 of FIG. 16B, the server proceeds to step 650, where the server 8 finds that what was received from the client CADD_c is encrypted data. In the next step 651, the server 8 reads a server secret key SK_s from an IC card 7. In the next step, the server 8 decrypts the received encrypted data with the server secret key SK_s as follows:

$$d1(SK_s, \text{encrypted data}) = d1(SK_s, e1(PK_s, R + e2(R, CCNOu))) = R + e2(R, CCNOu).$$

In step 653, the server 8 makes a check to see if the just obtained pseudo random number R coincides with the random number R which has been stored in a memory location R' of the server. If so, the server 8 sends an enable message to the client of CADD_c, and in step 655 decrypts e2(R, CCNOu) with the pseudo random number R to obtain the user's credit card number CCNOu. On the other hand, in response to a reception of the enable message in step 657, the client 2 exits from the process. After step 655, the server also exits from the process. If the result is NO in step 653, then the server 8 sends a disable message to the client in step 656, and ends the process. In response to a reception of the disable message in step 657, then the client displays a message to this effect in step 658, and then ends the process.

After operation of block 650, the client 2 waits, in step 663, for a report from the server on whether the credit card for the transmitted card number (CCNOu) is valid or not, while the server 8 refers to the credit company associated with the card number in step 661 to see if the credit card is valid. If not, the server 8 informs the client 2 of the invalidity of the credit card in step 662, and ends the process. If the card is valid in step 661, the server 8 informs the client of the validity in step 667. If the client 2 receives a report from the server in step 663, the client makes another check in step 664 to see if the report indicates the validity of the card. If not, the client displays a message to indicate the invalidity in step 665, and ends the process. If the report indicates the validity in step 664, which means the completion of step 667, then the client 2 and the server 8 proceed to the next block 670.

In step 670, the client 2 and the server 8 execute timed play and metered charge report routine. FIGs. 18A and 18B are flow charts jointly showing a procedure formed of routines 675ac and 675bc interactively executed for playing an application while timing the duration and displaying a charge and a total amount of charges after the play. In FIG. 18, the routines 675ac and 675bc are identical to the routine 675a and 675b in FIGs. 11A and 11B except that "time" has been replaced with "charge", and accordingly VM-METER and AM-METER have been replaced with VC-METER and AC-METER.

The operation, in the client 2, of playing an application on usage-sensitive charging is completed by block 675 of FIG. 15 or step 218a of FIG. 18A. After step 212a, the server 8 charges the play to the credit card number CCNOu obtained in step 655 of FIG. 17B in step 680. This completes the whole of the charged application play process of FIG. 15.

In this process, only information on charge is given to the user. It is very easy to provide information on both time and charge by adding steps 91 through 93 and 95 to the routines 640b and 640a, and by adding steps 210 and 218 to the routines 675bc and 675ac.

As described above, expected time and/or charge are (is) displayed before playing a user specified application. This is helpful for the user to decide whether to play the application. Additionally, charging is done based on the actually timed play duration. This makes the charging reasonable.

In the above description, the arrangement is such that the user has to input his or her credit card number CCNOu each time he or she wants to play an application. However, instead of doing this, the credit card number CCNOu may be stored in non-volatile memory or EEPROM 103 in a PW_u-encrypted form. In this case, CCNOu is obtained by decrypting PW_u-encrypted CCNOu (e.g., e(PW_u, CCNOu)) with a password entered by the user. That is, d(entered password, e(PW_u, CCNOu)) = CCNOu.

Permit the Play Within a Preset Limit

FIG. 19 is a flow chart showing a procedure interactively executed by the client 2 and the server 8 in the operation block 800 of FIG. 5, wherein blocks connected with two flow lines indicates that operation of the blocks is done by the two elements 2 and 8. In this case, it is assumed that a preset limit is recorded in or on the application package and is transmitted from client 2 to server each time of play. On entering the process 800 via step 516 of FIG. 5, the client 2 proceeds to step 801, where the client 2 and the server 8 executes the initial routines 80. It is noted that in routine 80b, if there is a record for VID_v and $NO_{v,t}$, then the limit value ($LV_{v,t}$) field of the table 60 of FIG. 6A contains the limit value transmitted from the client 2, otherwise, the received limit value is stored in the $LV_{v,t}$ field when the record for VID_v and $NO_{v,t}$ is added in step 88.

In step 810, the server 8 makes a check if a meter associated with the TOU code received from the client 2 is under the limit value. This check is made by comparing an LV field and LV-meter field associated with the TOU code in table 60. If the value of the LV-meter is equal to or greater than the LV field value, then the server returns an over limit message to the client 2 in step 820. If not, the server 8 returns an underlimit message to the client 2 in step 822, and proceeds to step 828. If the client 2 receives the overlimit message in step 824, then the client 2 displays a message to this effect. If not, the client 2 proceeds to the step 828.

Since the expected play time informing routines 97a and 97b and the application play subroutine 600 has been described above, the description of steps 828 and 830 are omitted.

According to this feature of the invention, it is possible to limit the use of charged information. This feature is especially useful in case when a user who have paid in advance for the use of the application package is permitted to use the application package within a limit value.

Though it has been assumed that the limit values are included in the application package, the limit values may be kept in the servers of the provider or distributor from the beginning. In this case, the limit values are fixed. However, if limit values are permitted to be set and recorded in the application package at the time of distribution or sales, the limit values are advantageously set according to an amount paid.

As is apparent from the foregoing, as a limit value, any use-limiting factors will do that can be measured in quantity. Such limit values are, for example, the effective date and time, the allowable expiration date and time, the maximum amount of play time, the allowable access count.

It is also possible to combine this feature with a charged application play feature. That is, an arrangement may be such that the user is permitted to use an application package on usage-sensitive charging only if the value of an LV-meter associated with the TOU is under the value of the corresponding LV or the value recorded in a field 33 or 34 of the distribution descriptor 23.

Modification I

In the above embodiment, applications, if more than one, in one volume are encrypted by an identical application encrypting key K_v . However, the applications AP_a in one volume may be encrypted with respective AP-encrypting keys K_a , where a lower case "a" following AP and K is a serial number assigned to each application ID. In this case, each of the AP-encrypting keys K_a are encrypted with the user public key PK_u , and stored in the PK_u -encrypted AP-encrypting key (K_a) fields 32a in the distribution descriptor 23.

Modification II

It has been assumed that the user of the DVD 3 is limited to the purchaser thereof who have had the PK_u -encrypted AP-encrypting key (K_v) recorded on the DVD 3. However, the system may be so arranged that predetermined people, e.g., family members FM_1, FM_2, \dots, FM_N of the purchaser can use the DVD (N is the number of the family members). One of the ways to realize this is to encrypt the AP-encrypting key K_v with a public key PK_{u-n} of each member FM_n ($n = 1, 2, \dots, N$) to obtain $e1(PK_{u-1}, K_v), e1(PK_{u-2}, K_v), \dots, e1(PK_{u-n}, K_v)$ and to record them in the PK_{u-n} -encrypted AP-encrypting key $e1(PK_{u-n}, K_v)$ fields 32 of the distribution descriptor 23 at the time of purchase of the DVD.

Modification III: K_v Retrieval From Server

In the above description, the AP-encrypting key K_v has been recorded in a PK_u -encrypted form on the DVD 3. However, the AP-encrypting key K_v may be managed by the server 8 and transmitted to the client or the DVD player 2 in response to a request issued from the DVD player 2 each time of use of the DVD 3. In this case, there is no need of providing the distribution descriptor 23 with the PK_u -encrypted AP-encrypting key field 32. Instead each of the servers has to store an AP-encrypting key table (or K_v table) and a PK_u table (shown in FIGs. 20A and 20B) in the hard disc. As shown in FIG. 20A, the K_v table a volume ID (VID_v) field (as the entry of record) and an AP-encrypting key (K_v) field in

each record. In FIG. 20B, each record of the PK_v table comprises a volume ID (VID_v) field (as the entry of record), a volume issue number ($NO_{v,i}$) field and a PK_v field (Successive same values in the first field are shown by showing only the first appearing one). Further, the process (or step) 610 of obtaining the AP-encrypting key K_v , that is, a group of the steps 602, 604 and 606 in the application play routine 600, has to be replaced with a process of FIG. 20C.

5 FIG. 20C is a flow chart of a process in which the client DVD player 2 obtains the application encrypting key K_v from the server 8. In step 616, the server 8 retrieves a key K_v from the K_v table by using VID_v . In the next step 618, the key K_v is encrypted with an arbitrary number used only in the current process, e.g., a pseudo random number R to obtain $e2(R, K_v)$. In the next step 620, the server 8 retrieves a key PK_v from the PK_v table by reading the PK_v field of the record which contains VID_v and $NO_{v,i}$ in the VID_v and $NO_{v,i}$ fields, respectively. In the next step 622, $R + e2(R, K_v)$ is encrypted
10 with the retrieved key PK_v to obtain a double encrypted AP-encrypting key
$$e1(PK_v, R + e2(R, K_v)),$$

which is returned to the client with a client network address $CADD_c$ in the next step 624.

On the other hand, the controller 100 of the client 2 waits for a response from the server 8 of SID_a in step 626. If there is any response from the server 8 of SID_a in step 626, then the client DVD 3 receives the data $e1(PK_v, R + e2(R, K_v))$ from the server 8 in step 628. In the next step 630, the received data is decrypted with the user secret key SK_u read
15 from the IC card 5. Specifically, the following calculation is done.

$$d1(SK_u, e1(PK_v, R + e2(R, K_v))) \Rightarrow R + e2(R, K_v)$$

In the next step 632, $e2(R, K_v)$ is decrypted with the obtained pseudo random number R . Specifically, the following calculation is done.

20
$$d2(R, e2(R, K_v)) \Rightarrow K_v$$

Thereafter, the controller 100 proceeds to the step 608 of FIG. 14.

In this modification, the applications AP_i in one volume may be encrypted with respective AP-encrypting keys K_a . In this case, the K_v table has to be replaced with K_a table in which each record comprises an application ID (AID_a) field and an AP-encrypting key (K_a) field. Further in step 612, the controller 100 of the DVD player 2 has to also send the application ID of the application to be played to the server.
25

Also in this modification, the system may be, again, so arranged that predetermined people, e.g., family members FM_1, FM_2, \dots, FM_N of the purchaser can use the DVD (N is the number of the family members). In this case, for each member FM_n ($n = 1, 2, \dots, N$), the server 8 has to use the member's own public key PK_{v+n} in encrypting the AP-encrypting key K_v . One way to realize this is to issue a volume issue number NO_{v+n} to each member FM_n at the time of sales of the DVD, provide the non-volatile memory (not shown) of the DVD player 2 with a table for associating the user's password PW_n with the volume issue number NO_{v+n} , send the volume issue number (NO_{v+n}) associated with the user's password in step 612, and use not the PK_v table but a PK_{v+n} table in which each of the records has the following fields:
30

$$VID_v, NO_{v+n}, PK_{v+n}$$

35 Another way is to issue and record not only a volume issue number $NO_{v,i}$ but also family member numbers FMN_n for all members at the time of sales of the DVD, provide the non-volatile memory (not shown) of the DVD player 2 with a table for associating the user's password PW_n with the corresponding family member number FMN_n , send the volume issue number ($NO_{v,i}$) and the family member number FMN_n associated with the user's password in step 612, and use another PK_{v+n} table in which each of the records has the following fields:
40

$$VID_v, NO_{v,i}, FMN_n, PK_{v+n}$$

In the process of FIG. 20C, the server 8 may be authenticated by means of a public-key cryptosystem using a pair of server secret and public keys (SK_a, PK_a). In this case, the server 8 signs the double-encrypted AP-encrypting key
45
$$e1(PK_v, R + e2(R, K_v))$$

with a signing key or the server secret key SK_a after step 622. While the client or DVD player 2 tests the signature by the server 8 with a test key or the server public key PK_a contained in the PK_a field 31 of the distribution descriptor 23 recorded in the burst cutting area of the DVD 2 before step 630.

However, even if just described authentication of the server 8 is omitted, an attacker will never go to any greater length than a steal of TOU code plus limit value, a volume ID VID_v , a volume issue number $NO_{v,i}$, and the client network address $CADD_c$. This is not a serious problem.

50 In the process of FIG. 20C, a pseudo random number R has been used as a pseudo variable which takes a different value each time of execution of the process. However, as the pseudo variable, any thing will do if the result of encryption with it takes a different value each time of execution of the process.

Modification IV

55 In the first illustrative embodiment, the decryption of application is achieved by software. For this purpose, the controller 100 has to read the user secret key SK_u from the IC card 5 through the bus 102, which leaves the possibility of permitting a breaker to easily steal the user secret key SK_u through the bus 102. In order to prevent this, the process

achieved by the steps 604 through 608 may be realized by hardware as shown in FIG. 21, which is a block diagram of an exemplary decipherer-built-in IC card IF. In FIG. 21, the decipherer-built-in IC card IF 120a comprises an IC card receptacle 121 and a printed wiring board 122 extending from and fixed with the receptacle 121. An IC 123 is mounted on the printed wiring board 122. The IC 123 comprises a memory IF 125 which usually connects the memory of the IC card 5 with the bus 102 and, in response to an instruction from the controller 100, reads and passes the key SK_u to the next stage; a K_v decoder 126 for receiving the key SK_u and encrypting $e1(PK_u, K_v)$ with the key SK_u to yield K_v ; and an AP decoder 127 for receiving the key K_v and encrypting $e(K_v, AP)$ to yield application data (AP). The printed wiring board 122 portion may be preferably molded together with the IC card receptacle 121 portion so as to make the whole a single body. By doing this, leaking of the user secret key SK_u can be prevented.

This modification can be also applied to a system 1 using the cryptosystem of FIG. 20C. In this case, the K_v decoder 126 of FIG. 21 has to be replaced with a K_v decoder 126a as shown in FIG. 22. In FIG. 22, the K_v decoder 126a decrypts the input data, $e1(PK_u, R + e2(R, K_v))$, from the bus 102 by using the user secret key SK_u passed by the memory IF 125 to obtain $R + e2(R, K_v)$, while decrypting the obtained data $e2(R, K_v)$ with the obtained random number R and outputting the key K_v .

Embodiment II

FIG. 24 is a block diagram showing an arrangement of a system capable of playing a distributed application package, e.g., a DVD on the terms of use of the DVD without communicating with any server according to a second illustrative embodiment of the invention. In FIG. 24, the system 1a is identical to the client 2 of FIG. 1 except that the communication IF 150 has been eliminated because of no need of communication with a server and the controller 100 has been replaced with a controller 100a. In the controller 100a, a not-shown ROM for storing a control program as described later and the EEPROM 103 have been also replaced with a new ROM (not shown) and an EEPROM 103a. In order to play a role of the server 8, the system 1a has to have table 60 of FIG. 6A in any non-volatile memory, e.g., the EEPROM 103a and an application duration (play time) for each application as defined in table 70 of FIG. 6B has to be included in the control data of each application package.

FIG. 25 schematically shows an exemplary control program executed by the controller 100a shown in FIG. 24. The control program of FIG. 25 is also identical to that of FIG. 5 except that the decision step 516 and the step 700 has been eliminated because the limit-attached play mode is not supported by the system 1a in this embodiment, and the steps 650 and 800 are replaced with steps 650a and 800a. Accordingly, operation after step 514 will be described in the following.

If the lower digit of the terms-of-use (TOU) code is 0 in the decision step 514, then in step 650a the controller 100a plays, in the free play mode, the application stored in the selected application in step 506 or 512 and ends the operation. It should be noted that since the system 1a does not have the charged play mode, the lower digit of the TOU code is defined as follows.

Higher digit of terms-of-use code (Hexadecimal)	Corresponding limit value	Play mode
0	None	Free play mode
2	Effective date and time	Limit-attached play mode
3	Allowable expiration date and time	
4	Maximum amount of used period	
5	Allowable access count	
:	:	
:	:	

Accordingly, if the lower digit of the TOU code is not 0 in the decision step 514, then in step 800a the controller 100a plays, in the limit-attached play mode, the application stored in the selected application in step 506 or 512 and ends the operation.

FIGs. 26 and 27 show an operation of a free play mode shown in step 650a of FIG. 25 in a detailed form and a further detailed form, respectively. In FIG. 26, the controller 100a executes an initial routine 80a in step 660a, in step 670a executes an expected play time informing routine, and in step 680a executes an application play and metered play time report routine.

As shown in FIG. 27, in the initial routine 80c, the controller 100a searches the table 60 for a record which contains VID_v and $NO_{v,i}$ in the volume ID and issue No. fields thereof, respectively in step 86. If the search is unsuccessful, then the controller 100a adds the record for VID_v and $NO_{v,i}$ and fills relevant fields with $AID_{v,i}$ and a limit value, if any, in the table 60 in step 88, and proceeds to step 90. Also, if the search in step 86 is successful, the server 9 proceeds to step 90, where the controller 100a selects a routine to execute next according to the value of the TOU code and enters the selected routine. In this case, if the TOU code = x0H (x: an arbitrary HEX number, the letter H in the last position indicates that the preceding number is in hexadecimal), then a routine for playing an application free of charge is selected. If the TOU code \geq x1H, then a routine is selected which plays an application only if the software meter of a use-limiting factor is under a preset value.

The expected play time informing routine 670a is identical to the routines 97 (FIG. 10) minus communication steps 93 and 94, comprising the above described steps 91, 92 and 95. Similarly, it is seen from FIGs. 11 and 13A that the above described steps 620, 622, 624, 210 and 218 are executed in this order in the timed play and metered usage report routine 680a. In this way, the system 1a permits the user to play the application stored in the selected application (steps 506 and 512 of FIG. 25) free of charge.

FIG. 28 is a flow chart showing an operation of a limit-attached play mode shown in step 800a of FIG. 25. Since this operation is very similar to that of FIG. 19, only the flow is briefly described, omitting the details of each step. In FIG. 28, controller 100a first makes a check if a meter associated with the TOU code has reached the limit value obtained with the TOU code. If so, then the server returns an overlimit message to controller 100a in step 820. Otherwise, the controller 100a proceeds to the expected play time informing routine 828a (= 670a), where the controller 100a executes the above described steps 91, 92 and 95, and then calls the application play subroutine 600 in step 830, thereby completing the operation. Since the application play subroutine 600 has been detailed above, further description is omitted. In this way, the system 1a permits the user to play the application stored in the selected application (steps 506 and 512 of FIG. 25) only if the limit value associated with the TOU code assigned to the volume or the user-specified application has not been reached.

According to the second embodiment, the system 1a can operate in either of the free play mode and the limit-attached play mode without the need of communication with a server. For this, the system 1a may be made portable.

Modifications

In the above description, the illustrative embodiment has been described in conjunction with the DVD. The same discussion can be applied to such package media as permit write once or more.

Further, the present invention is also applicable to application packages distributed via transmission media. In this case, the distributed application packages are stored in a bulk storage in the user's device. An application package comprises one or more application and application control data, that is, an application descriptor and distribution descriptor. One volume is stored as a file. Since a plurality of application package may be stored in a single storage, each application package does not have to contain a control program. One control program, which may be distributed via either package or transmission media, is enough for one user device. The folder or directory in which the application packages are stored is set for a user specified one in the control program when the control program is installed. The data to be recorded in the distribution descriptor is included in the application package by the provider according to the information given by the user.

As described above, one who is permitted to use an application package is limited to an owner of the IC card which stores a user secret key SK_u corresponding to the user public key PK_u used for encryption of the AP-encrypting key K_v in the application package. For this, even if someone has unjustly obtained an application package, for example, by copying the whole volume from the DVD on which the volume is recorded, he or she can not use it without the IC card of the owner of the DVD. Thus the inventive system can prevent unjust use of an application package (DVD in this case) by any other person than the regular owner of the application package.

Also, the inventive system is so arranged that most part of the application package is recorded by pressing in manufacturing process of the DVDs, whereas at least a part of the volume control data (i.e., the distribution descriptor) can be determined at the time of, e.g., distribution of each of the DVDs after the manufacturing process. This makes the system flexible because control data can be easily changed without changing the stamper.

In the initial routines 80a and 80b in FIG. 8A and 8B, the data transmitted with the service request may be encrypted in the same manner as in case of the transmission of user's credit card number shown in FIG. 17. However, in case of the initial routines, there are a plurality of data. These data may be encrypted in the following way.

If the data to be encrypted are $D1, D2, \dots$ then they are first encrypted with a key R as follows:

$e2(R, D1), e2(R, D2), \dots$

Then further encryption is made with a server public key PK_s as follows:

$e1(PK_s, R + e2(R, D1) + e2(R, D2), \dots)$.

In the process of FIG. 17, the user may be authenticated by means of a public-key cryptosystem using a pair of

user secret and public keys (SK_u , PK_u). In this case, the client 2 signs the double-encrypted credit card number $e1(PK_s, R + e2(R, CCNOu))$ with a signing key or the user secret key SK_u after step 648. While the server tests the signature by the client 2 with a test key or the user public key PK_u before step 650.

5 Instead of storing a single server public key in the distribution descriptor 23, a plurality of server public keys or all the server public keys may be recorded. By doing this, it is possible, for example, to setting a different charge depending on the server public key which the user have selected by appropriately combining the tables 70 and 75.

Also, application packages with an identical volume ID can have different server public keys recorded. A plurality of toll center may be advantageously provided for application packages of the same title.

10 In order to prevent any use of IC card by other person than the owner of the IC card, it is possible to add, before the SK_u reading step 604, the steps of prompting the user to enter a password through a human IF 110 and proceeding to step 604 only if the entered password coincides with the user password PW_u stored in the IC card.

Though the IC card 5 is used in the above embodiment, the IC card IF 120 may be replaced with a magnetic card reader to permitting the use of the magnetic card. Alternatively, the arrangement may be such that the user enters his or her password each time the user uses the DVD.

15 Instead of storing the user secret key SK_u in the IC card 5, the key SK_u may be stored in non-volatile memory in a PW_u -encrypted form. In this case, the key SK_u is obtained by decrypting PW_u -encrypted SK_u with a password entered by the user.

The discussion of three preceding paragraphs are applied to the IC card used for storing the server secret key in the server. However, in this case the user has to be taken as the administrator of the toll server.

20 Many widely different embodiments of the present invention may be constructed without departing from the spirit and scope of the present invention. It should be understood that the present invention is not limited to the specific embodiment described in the specification, except as defined in the appended claims.

A system for permitting only an authentic user to play a desired application contained in a distributed application package in one of predetermined operation, e.g., free play mode, charged mode, limit-attached play mode, etc. The system comprises a client for playing an application under the control of a server connected with the client through a communication network. The application package (the volume) includes a distribution descriptor which contains mode codes assigned to the volume and the applications of the volume. The data of distribution descriptor is decided and stored in the descriptor at the time of distribution of the volume. This feature makes the system flexible. There is also disclosed a system operatable without communicating with a server.

30

Claims

1. An application package for use in a system for playing an application contained in the application package (the volume), the application package comprising:

35

application data for at least one application; and
 volume control data for use in controlling said system, wherein said volume control data at least comprises:
 a volume ID for identifying the kind of said application package (said volume);
 40 an issue number assigned in order of issue to each of the volumes of said kind; and
 application IDs each assigned to one of said at least one application contained in said volume, and wherein:
 at least a part of said volume control data is to be added to said volume after the creation of said volume; and
 said at least a part of said volume control data includes said issue number.

45 2. An application package as defined in claim 1, wherein:

said application data has been encrypted with an encrypting key; and
 said at least a part of said volume control data includes a user's public key-encrypted version of said encrypting key used.

50

3. An application package as defined in claim 1, wherein said at least a part of said volume control data includes mode codes which are assigned to said volume or said at least one application and each indicate a play mode associated with one of said volume or said at least one application to which the mode code is assigned.

55 4. A package media on which an application package as defined in claim 1 has been recorded.

5. A package media of a write-once type on which an application package as defined in claim 1 has been recorded.

6. A package media on which an application package as defined in claim 1 has been recorded wherein said at least a part of said volume control data is recorded in an area different from data area where said application data is recorded on the package media.
- 5 7. A method for sending data with a raised security from a first device to a second device through a public telecommunication network, comprising the steps of:
- in said second device,
- 10 generating a pseudo random number;
 transmitting said pseudo random number to said first device;
- in said first device,
- 15 encrypting said data with said transmitted pseudo random number into encrypted data;
 encrypting concatenated data consisting of said pseudo random number and said encrypted data with a public key of said second device into double-encrypted data;
 sending said double-encrypted data to said second device; in said second device,
 decrypting said double-encrypted data with a secret key of said second device which corresponds to said
 20 public key into decrypted data consisting of a decrypted random number portion and another decrypted portion; and
 decrypting said another decrypted portion with said transmitted random number to obtain said data.
8. A method for sending a plurality of pieces of data with a raised security from a first device to a second device through a public telecommunication network, comprising the steps of:
- 25 in said second device,
- generating a pseudo random number;
 transmitting said pseudo random number to said first device;
- 30 in said first device,
- encrypting each of said pieces of data with said transmitted pseudo random number into an encrypted
 35 piece of data;
 encrypting concatenated data consisting of said pseudo random number and said encrypted pieces of data with a public key of said second device into double-encrypted data;
 sending said double-encrypted data to said second device; in said second device,
 decrypting said double-encrypted data with a secret key of said second device which corresponds to said
 40 public key into decrypted data consisting of a decrypted random number portion and said plurality of decrypted data portions; and
 decrypting each of said decrypted portions with said transmitted random number to obtain said pieces of data.
- 45 9. A method as defined in claim 7 or 8, further comprising the steps, executed after said step of decrypting said double-encrypted data, of:
- proceeding to a next step only if said decrypted random number portion coincides with said transmitted pseudo
 50 random number; and
 said second device informing said first device of a failure in decryption if said decrypted random number portion does not coincide with said transmitted pseudo random number.
10. In a system provided with means for playing an application contained in an application package, a method for permitting a user to play an encrypting key-encrypted application contained in a distributed application package which further contains, as volume control data, a user's public key-encrypted encrypting key so encrypted as to be able to be decrypted with a secret key of the user into said encrypting key, the method comprising the steps of:
- 55 reading said user's public key-encrypted encrypting key from said distributed application package (said vol-

ume);

obtaining said secret key;

decrypting said user's public key-encrypted encrypting key with said secret key to obtain said encrypting key;
and

6 decrypting said encrypting key-encrypted application with said obtained encrypting key into application data while passing said application data to said means for playing an application.

11. In a system comprising a client provided with means for playing an application contained in an application package and a server connected with the client through a communication network, a method for permitting a user to play
10 one of encrypting key-encrypted applications contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order and application IDs, the method comprising the steps of:

16 said client reading said volume ID, said issue number and an application ID for said one of encrypting key-encrypted applications (said encrypting key-encrypted application) from said volume and sending to said server;

in said server,

20

retrieving said encrypting key by using said volume ID;

retrieving a public key of said user by using said volume ID and said issue number;

generating a pseudo random number;

25

double-encrypting said encrypting key with said pseudo random number and said public key into a double encrypted data;

sending said double-encrypted data to said client; in said client,

obtaining a secret key of said user which corresponds to said public key;

obtaining said encrypting key by decrypting said double-encrypted data with said secret key;

30

decrypting said encrypting key-encrypted application with said obtained encrypting key into application data while passing said application data to said means for playing an application.

12. A method as defined in claim 10 or 11, wherein said means for obtaining a secret key comprises means for reading said secret key from a portable memory of said user.

35

13. A method as defined in claim 12, wherein said portable memory is an IC card.

14. In a system comprising a client provided with means for playing an application package and a server connected with the client through a communication network for controlling the client, the application package (the volume) containing, as volume control data, a volume ID and an issue number issued to each of the volumes of said volume ID
40 in an issued order, a method for controlling the amount of play time comprising the steps of:

said client sending said volume ID and said issue number to said server;

said server retrieving an expected play time associated with said volume ID and said issue number; and

45

said server adding said expected play time to the value of a total play time associated with said volume ID and said issue number.

15. In a system comprising a client provided with means for playing an application contained in an application package and a server connected with the client through a communication network for controlling the client, the application package (the volume) containing, as volume control data, a volume ID, an issue number issued to each of the volumes of said volume ID in an issued order and an application ID for the application, a method for controlling the amount of play time comprising the steps of:

50

said client sending said volume ID, said issue number and said application ID to said server;

said server retrieving an expected play time associated with said volume ID, said issue number and said application ID; and

55

said server adding said expected play time to the value of a total play time associated with said volume ID and said issue number.

16. In a system comprising a client provided with means for playing an application contained in an application package and a server connected with the client through a communication network for controlling the client, the application package (the volume) containing, as volume control data, a volume ID and an issue number issued to each of the volumes of said volume ID in an issued order, a method for controlling the amount of play time comprising the steps of:
- 5
- said client and said server interactively measuring, as a measured play time, a play time of said application;
 - and
 - said server adding said measured play time to the value of a total play time associated with said volume ID and
- 10
- said issue number.
17. A method as defined in claim 16, wherein said step of measuring a play time comprises the step of using a timer of said server.
18. A method as defined in claim 16, wherein said step of measuring a play time comprises the step of using a timer of said client.
19. In a system comprising a client for playing an application package and a server connected with the client through a communication network wherein the application package (the volume) comprises application data and control data and at least a part of the control data has been added to the volume after the creation of said volume, a method for sending desired data from one side of said client and said server to the other side, the method comprising the steps of:
- 20
- including a secret key of said other side in said at least a part of said control data;
- 25
- in said other side,
 - generating a pseudo random number;
 - transmitting said pseudo random number to said one side;
- 30
- in said one side,
 - encrypting said desired data with said transmitted pseudo random number into encrypted data;
 - encrypting concatenated data consisting of said pseudo random number and said encrypted data with said public key of said other side into double-encrypted data;
 - sending said double-encrypted data to said other side;
- 35
- in said other side,
 - decrypting said double-encrypted data with a secret key of said other side which corresponds to said public key into decrypted data consisting of a decrypted random number portion and another decrypted portion; and
 - decrypting said another decrypted portion with said transmitted random number to obtain said desired data.
- 40
- 45
20. A method as defined in claim 19, wherein said generating a pseudo random number includes storing said pseudo random number in memory, and wherein the method further comprises the step, executed prior to said decrypting said another decrypted portion, of:
- 50
- in response to a determination that said decrypted random number portion does not coincide with said pseudo random number stored in said means for storing said pseudo random number stored in said memory, informing said one side of a failure in decryption instead of passing the control to next means.
- 55
21. In a system comprising a client provided with means for playing an application contained in an application package and a server connected with the client through a communication network, a method for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order, and an application ID for said application, the method comprising the steps of:

proceeding to a next step only if the value of a meter field associated with said volume ID, said issue number and said application ID is under the value of a limit value field associated with said volume ID, said issue number and said application ID in a volume data table; and displaying a message informing an overlimit on a display device of said client and quit the operation otherwise.

5

22. In a system comprising a client provided with means for playing an application contained in an application package and a server connected with the client through a communication network, a method for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order, an application ID for said application and a limit value for limiting the play of said application, the method comprising the steps of:

10

proceeding to a next step only if the value of a meter field associated with said volume ID, said issue number and said application ID in a volume data table is under said limit value; and displaying a message informing an overlimit on a display device of said client and quit the operation otherwise.

15

23. A method as defined in claim 21, wherein said limit value is one of effective date and time, allowable expiration date and time, a maximum amount of play time, and an allowable access count.

24. A method as defined in any of claims 11, 15 and 16, wherein said step of said client sending to said server comprises the steps of:

20

said client encrypting at least one of said volume ID, said issue number and said application ID into encrypted data; and said server decrypting said encrypted data.

25

25. A system for sending data with a raised security from a first device to a second device through a public telecommunication network, comprising:

30

means provided in said second device for generating a pseudo random number;
 means provided in said second device for transmitting said pseudo random number to said first device;
 means provided in said first device for encrypting said data with said transmitted pseudo random number into an encrypted data;
 means provided in said first device for encrypting concatenated data consisting of said pseudo random number and said encrypted data with a public key of said second device into double-encrypted data;
 means provided in said first device for sending said double-encrypted data to said second device;
 means provided in said second device for decrypting said double-encrypted data with a secret key of said second device which corresponds to said public key into decrypted data consisting of a decrypted random number portion and another decrypted portion; and
 means provided in said second device for decrypting said another decrypted portion with said transmitted random number to obtain said data.

35

40

26. A system for sending a plurality of pieces of data with a raised security from a first device to a second device through a public telecommunication network, comprising:

45

means provided in said second device for generating a pseudo random number;
 means provided in said second device for transmitting said pseudo random number to said first device;
 means provided in said first device for encrypting each of said pieces of data with said transmitted pseudo random number into an encrypted piece of data;
 means provided in said first device for encrypting concatenated data consisting of said pseudo random number and said encrypted pieces of data with a public key of said second device into double-encrypted data;
 means provided in said first device for sending said double-encrypted data to said second device;
 means provided in said second device for decrypting said double-encrypted data with a secret key of said second device which corresponds to said public key into decrypted data consisting of a decrypted random number portion and said plurality of decrypted data portions; and
 means provided in said second device for decrypting each of said decrypted portions with said transmitted random number to obtain said pieces of data.

50

55

27. A system as defined in claim 25 or 26, further comprising:

5 means, provided in said second device, activated prior to decrypting each of said decrypted portions and responsive to a determination that said decrypted random number portion does not coincide with said transmitted pseudo random number, for informing said first device of a failure in decryption instead of passing the control to next means.

28. A system for playing an encrypting key-encrypted application contained in a distributed application package which further contains, as volume control data, a user's public key-encrypted encrypting key so encrypted as to be able
10 to be decrypted with a secret key of the user into said encrypting key, the system comprising:

means for reading said user's public key-encrypted encrypting key from said distributed application package (said volume);
means for obtaining said secret key;
15 means for decrypting said user's public key-encrypted encrypting key with said secret key to obtain said encrypting key;
means for decrypting said encrypting key-encrypted application with said obtained encrypting key to provide application data; and
means for using said application data for playing.
20

29. A system for permitting a user to play an encrypting key-encrypted application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order and application IDs, the system comprising:

25 a client for playing an application by using application data; and
a server for controlling said client through a communication network, wherein said client comprises:
means for reading and sending said volume ID, said issue number and an application ID for said one of encrypting key-encrypted applications (said encrypting key-encrypted application) from said volume to said
30 server, said server comprises:

means for retrieving said encrypting key by using said volume ID;
means for retrieving a public key of said user by using said volume ID and said issue number;
means for generating a pseudo random number;
35 means for double-encrypting said encrypting key with said pseudo random number and said public key into a double encrypted data; and
means for sending said double-encrypted data to said client, and said client comprises:
means for obtaining a secret key of said user which corresponds to said public key;
means for obtaining said encrypting key by decrypting said double-encrypted data with said secret key;
40 means for decrypting said encrypting key-encrypted application with said obtained encrypting key to provide application data; and
means for using said application data for playing.

30. A system as defined in claim 28 or 29, wherein said means for obtaining a secret key comprises means for reading
45 said secret key from a portable memory of said user.

31. A system as defined in claim 30, wherein said portable memory is an IC card.

32. A system for permitting a user to play a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume) and an issue number
50 issued to each volume of the kind in an issued order, the system comprising:

a client for playing said distributed application package; and
a server for controlling said client through a communication network, wherein:
said client comprises means for sending said volume ID and said issue number to said server; and
said server comprises means for retrieving an expected play time associated with said volume ID and said
55 issue number, and means for adding said expected play time to the value of a total play time associated with said volume ID and said issue number.

33. A system for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order and an application ID for the application, the system comprising:

5

a client for playing said application; and
 a server for controlling said client through a communication network, wherein:
 said client comprises means for sending said volume ID, said issue number and said application ID to said server; and

10

said server comprises means for retrieving an expected play time associated with said volume ID, said issue number and said application ID, and means for adding said expected play time to the value of a total play time associated with said volume ID and said issue number.

34. A system for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order and an application ID for the application, the system comprising:

15

a client for playing said application; and
 a server for controlling said client through a communication network, wherein:
 said client and said server comprise means for interactively measuring, as a measured play time, a play time of said application; and
 said server further comprises means for adding said measured play time to the value of a total play time associated with said volume ID and said issue number.

20

35. A system as defined in claim 34, wherein said means for interactively measuring a play time comprises means for using a timer of said server.

25

36. A system as defined in claim 34, wherein said means for interactively measuring a play time comprises means for using a timer of said client.

30

37. A system for permitting a user to play an application package (the volume) comprising application data and control data wherein at least a part of the control data has been added to the volume after the creation of said volume, the system comprising:

35

a client for playing said volume; and
 a server for controlling said client through a communication network, wherein said server comprises means for storing a secret key of said server and said at least a part of said control data includes a public key corresponding to said secret key, and wherein the system comprises:

40

means provided in said server for generating a pseudo random number;
 means for storing said pseudo random number;
 means provided in said server for transmitting said pseudo random number to said client;
 means provided in said client for encrypting desired data with said transmitted pseudo random number into encrypted data;

45

means provided in said client for encrypting concatenated data consisting of said pseudo random number and said encrypted data with said public key into double-encrypted data;
 means provided in said client for sending said double-encrypted data to said server;
 means provided in said server for decrypting said double-encrypted data with said secret key into decrypted data consisting of a decrypted random number portion and another decrypted portion; and

50

means provided in said server for decrypting said another decrypted portion with said transmitted random number to obtain said desired data.

38. A system as defined in claim 37, further comprising:

55

means, provided in said server, activated prior to said decrypting said another decrypted portion and responsive to a determination that said decrypted random number portion does not coincide with said pseudo random number stored in said means for storing said pseudo random number, for informing said client of a failure in decryption instead of passing the control to next means.

39. A system for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order and application IDs, the system comprising:
- 5
- a client for playing an application by using application data; and
 - a server for controlling said client through a communication network, wherein said client comprises:
 - means for reading and sending said volume ID, said issue number and an application ID for said one of encrypting key-encrypted applications (said encrypting key-encrypted application) from said volume to said
 - 10 server, said server comprises:
 - means for proceeding to next step only if the value of a meter field associated with said volume ID, said issue number and said application ID is under the value of a limit value field associated with said volume ID, said issue number and said application ID in a volume data table; and
 - means for causing said client to display a message informing an overlimit on a display device of said client and
 - 15 quit the operation otherwise.
40. A system for permitting a user to play an application contained in a distributed application package which further contains, as volume control data, a volume ID for identifying the kind of said distributed application package (said volume), an issue number issued to each volume of the kind in an issued order, application IDs and limit values
- 20 associated with respective application IDs for limiting the play of respective applications, the system comprising:
- a client for playing an application by using application data; and
 - a server for controlling said client through a communication network, wherein said client comprises:
 - means for reading and sending said volume ID, said issue number, an application ID for said one of encrypting
 - 25 key-encrypted applications (said encrypting key-encrypted application) and a limit value associated with said application ID from said volume to said server, and wherein said server comprises:
 - means for proceeding to a next step only if the value of a meter field associated with said volume ID, said issue number and said application ID in a volume data table is under said limit value; and
 - means for causing said client to display a message informing an overlimit on a display device of said client and
 - 30 quit the operation otherwise.
41. A system as defined in claim 39, wherein said limit value is one of effective date and time, allowable expiration date and time, a maximum amount of play time, and an allowable access count.
- 35 42. A system as defined in any of claims 29, 33 and 34, wherein said means for sending to said server comprises means for encrypting at least one of said volume ID, said issue number and said application ID.
43. A method for permitting an authentic user to play a desired one of the applications contained in a distributed application package in a system capable of playing an application, wherein said application package (said volume) contains volume control data including mode codes assigned to said volume and the applications of said volume, the
- 40 method comprising the steps of:
- deciding to use one of predetermined play modes specified by one of said mode codes associated with said
 - 45 desired application; and
 - playing said desired application in said specified play mode.
44. A method as defined in claim 43, wherein the method further comprises the step of including, in said mode codes, values indicative of a free play mode and at least one limit-attached play mode which correspond(s) to respective
- 50 limit value(s) used for limiting usage.
45. A method as defined in claim 44, wherein said step of playing said desired application comprises the step of:
- in response to a determination that said one of said mode codes associated with said desired application
 - 55 includes a value indicative of said free play mode, simply playing said desired application.
46. A method as defined in claim 44, wherein said step of playing said desired application comprises the step of:
- in response to a determination that said one of said mode codes associated with the desired application

includes one of values indicative of said at least one limit-attached play mode, displaying a message to the effect that a limit value associated with said one of values has been reached instead of playing said desired application if said limit value has been reached.

5 47. A method as defined in claim 43, wherein said volume control data further includes a volume ID, an issue number and an application ID for each of said applications, and wherein said step of deciding to use one of predetermined play modes comprises the steps of:

10 obtaining said one of said mode codes associated with said desired application and corresponding limit value by using said application ID; and
 comparing said one of said mode codes with a meter value associated with said volume ID, said issue number and said application ID.

15 48. A method as defined in claim 45, wherein each of said applications has been each encrypted with an encrypting key and said volume control data includes a user's public key-encrypted version of said encrypting key (a public key-encrypted version encrypting key), and wherein said step of simply playing said desired application comprises the steps of:

20 reading said user's public key-encrypted encrypting key from said volume;
 obtaining a user's secret key which corresponds to said user's public key;
 decrypting said user's public key-encrypted encrypting key with said user's secret key to obtain said encrypting key; and
 decrypting said desired application with said obtained encrypting key.

25 49. A system for permitting an authentic user to play a desired one of the applications contained in a distributed application package, wherein said application package (said volume) contains volume control data including mode codes assigned to said volume and the applications of said volume, the system comprising:

30 means for deciding to use one of predetermined play modes specified by one of said mode codes associated with said desired application; and
 means for playing said desired application in said specified play mode.

35 50. A system as defined in claim 49, wherein the system further comprises means for including, in said mode codes, values indicative of a free play mode and at least one limit-attached play mode which correspond(s) to respective limit value(s) used for limiting usage.

51. A system as defined in claim 50, wherein said means for playing said desired application comprises:

40 means, responsive to a determination that said one of said mode codes associated with said desired application includes a value indicative of said free play mode, for simply playing said desired application.

52. A system as defined in claim 50, wherein said means for playing said desired application comprises:

45 means, responsive to a determination that said one of said mode codes associated with the desired application includes one of values indicative of said at least one limit-attached play mode, for displaying a message to the effect that a limit value associated with said one of values has been reached instead of playing said desired application if said limit value has been reached.

50 53. A system as defined in claim 49, wherein said volume control data further includes a volume ID, an issue number and an application ID for each of said applications, and wherein said means for deciding to use one of predetermined play modes comprises:

55 means for obtaining said one of said mode codes associated with said desired application and corresponding limit value by using said application ID; and
 means for comparing said one of said mode codes with a meter value associated with said volume ID, said issue number and said application ID.

54. A system as defined in claim 51, wherein each of said applications has been encrypted with an encrypting key and

said volume control data includes a user's public key-encrypted version of said encrypting key (a public key-encrypted version encrypting key), and wherein said means for simply playing said desired application comprises:

- 5 means for reading said user's public key-encrypted encrypting key from said volume;
- means for obtaining a user's secret key which corresponds to said user's public key;
- means for decrypting said user's public key-encrypted encrypting key with said user's secret key to obtain said encrypting key; and
- means for decrypting said desired application with said obtained encrypting key.

10 55. A method for permitting an authentic user to play a desired one of the applications contained in a distributed application package in a system comprising a client capable of playing an application and a server connected with said client through a communication network, wherein said application package (hereinafter referred to as "said volume") contains volume control data including mode codes assigned to said volume and the applications of said volume, the method comprising the steps of:

- 15 said client deciding to use one of predetermined play modes specified by one of said mode codes associated with said desired application; and
- playing said desired application in said specified play mode by means of cooperation between said client and said server.

20 56. A method as defined in claim 55, wherein the method further comprises the step of including, in each of said mode code, a value indicative of one of a free play mode, a charged play mode and at least one limit-attached play mode, wherein said volume control data further comprises a limit value associated with each of said at least one limit-attached play mode.

25 57. A method as defined in claim 55 or 56, wherein said volume control data further includes a volume ID, an issue number, and an application ID for each of said applications, and wherein said step of playing said desired application in said specified play mode includes an application play step of simply playing said specified application.

30 58. A method as defined in claim 57, wherein each of said applications contained in a distributed application package has been encrypted with an encrypting key and said volume control data includes a user's public key-encrypted version of said encrypting key (a public key-encrypted version encrypting key), and wherein said application play step comprising the steps of:

- 35 reading said user's public key-encrypted encrypting key from said volume;
- obtaining a user's secret key which corresponds to said user's public key;
- decrypting said user's public key-encrypted encrypting key with said user's secret key to obtain said encrypting key; and
- 40 decrypting said desired application with said obtained encrypting key.

45 59. A method as defined in claim 57, wherein each of said applications contained in a distributed application package has been encrypted with an encrypting key and said volume control data includes a user's public key-encrypted version of said encrypting key (a public key-encrypted version encrypting key), and wherein said application play step comprises the steps of:

- in said server,
- retrieving an encrypting key by using said volume ID;
- retrieving a user's public key associated with said volume ID and said issue number;
- 50 double-encrypting said encrypting key with a pseudo random number and said user's public key into a double encrypted data;
- sending said double-encrypted data to said client; in said client,
- obtaining a user's secret key which corresponds to said user's public key;
- obtaining said encrypting key by decrypting said double-encrypted data with said user's secret key;
- 55 decrypting said desired application with said obtained encrypting key.

60. A method as defined in claim 57, wherein said step of playing said desired application further comprises the steps, executed prior to said application play step, of:

said server retrieving an expected play time associated with said desired application; and displaying said expected play time on a display device of said client.

5 61. A method as defined in claim 57, wherein said step of playing said desired application further comprises the steps of:

measuring, as a measured play time, a duration of said application play step;
adding said measured play time to a play time meter associated with said mode code to obtain a total amount of play time; and
10 displaying said measured play time and said total amount of play time on a display device of said client after said application play step.

15 62. A method as defined in claim 61, wherein said step of measuring a duration comprises the step of measuring said play time by using a timer of said server.

63. A method as defined in claim 61, wherein said step of measuring a duration comprises the step of measuring said play time using a timer of said client.

20 64. A method as defined in claim 57, wherein said step of deciding to use one of predetermined play modes comprises deciding to use said charged play mode if said one of said mode codes associated with said desired application includes a value indicative of said charged play mode, and wherein said step of playing said desired application comprises the steps of:

said client obtaining and sending a credit card number of said user to said server;
25 proceeding to a next step only if the credit card of said number is found to be valid from a reference to an associated credit company;
displaying, on a display device of said client, a charge for play decided based on a measurement of a duration of said application play step and a total amount of play charges after said application play step; and
said server charging said play to said credit card number.

30 65. A method as defined in claim 64, wherein said step of playing said desired application further comprises the steps, prior to said application play step, of:

35 displaying, prior to said application play step, an expected charge and an expected total amount of charges on said display device; and
letting the user decide whether to play said desired application.

40 66. A method as defined in claim 64, wherein said step of said client obtaining and sending a credit card number of said user to said server comprises the steps of:

in said server,

45 generating a pseudo random number;
storing said pseudo random number in memory;
transmitting said pseudo random number to said client;

in said client,

50 prompting said user to input said credit card number;
double-encrypting said credit card number first with said transmitted random number and then with a server's public key included in said volume control data into a double-encrypted number;
sending said double-encrypted number to said server; in said server,
decrypting said double-encrypted number with a server's secret key into a decrypted random number and another decrypted data; and
55 decrypting said another decrypted data with said transmitted random number to obtain said credit card number.

67. A method as defined in claim 66, wherein said step of said client obtaining and sending a credit card number of said

user to said server further comprises the steps, executed prior to said step of decrypting said another encrypted data, of:

5 proceeding to a next step only if said decrypted random number coincides with said pseudo random number which has been stored in said memory; and
displaying a message informing a failure in decryption and quitting the operation otherwise.

68. A method as defined in claim 57 wherein said step of deciding to use one of predetermined play modes comprises deciding to use one of said at least one limit-attached play mode if said one of said mode codes associated with
10 said desired application includes a value indicative of said one of said at least one limit-attached play mode, and wherein said step of playing said desired application comprises the step of:

15 in response to a determination that a meter value associated with said one of said mode codes associated with said desired application in a record identified by said volume ID, said issue number and an application ID of said desired application in a volume data table has reached a limit value associated with said mode code, displaying a message informing an overlimit on a display device of said client instead of executing said application play step.

69. A method as defined in claim 68, wherein said limit value is one of effective date and time, allowable expiration date and time, a maximum amount of play time, and an allowable access count.

70. A system for playing a distributed application package in one of predetermined play modes in concert with a server, wherein the application package contains a data set encrypted with an encrypting key (a K-encrypted data set) for each of at least one application and volume control data for use in controlling operation of the system and the
25 server and the volume control data includes mode codes defining said play modes, the system comprising:

means for permitting a user to select one of said at least one application of said volume;
means for deciding to use one of said predetermined play modes associated with one of said mode codes assigned to said selected application; and
30 means for playing said selected application in said selected play mode in concert with said server.

71. A system as defined in claim 70, wherein each of said mode codes includes one of values for a free play mode, a charged play mode and at least one limit-attached play mode.

35 72. A system as defined in claim 70, wherein said volume control data further includes a volume ID, an issue number and an application ID for each of said applications, and wherein said means for playing said selected application in said selected play mode at least comprises:

40 means for setting said server for said selected play mode by sending to said server said volume ID, said issue number, and the application ID and said mode code associated with said selected application; and
application play means for simply playing said specified application.

73. A system as defined in claim 72, wherein said volume control data further includes a user's public key-encrypted encrypting key, and wherein said application play means comprises:

45 means for reading said user's public key-encrypted encrypting key from said volume;
means for obtaining a user's secret key which corresponds to said user's public key;
means for decrypting said user's public key-encrypted encrypting key with said user's secret key to obtain said encrypting key; and
50 means for decrypting the K-encrypted data set of said selected application with said obtained encrypting key.

74. A system as defined in claim 73, wherein means for decrypting said user's public key-encrypted encrypting key and said means for decrypting the K-encrypted data set are realized as an integrated circuit.

55 75. A system as defined in claim 72, wherein said application play means comprises:

means for receiving double-encrypted data from said server;
means for obtaining a user's secret key which corresponds to said user's public key;

means for obtaining said encrypting key by decrypting said double-encrypted data with said user's secret key;
and

means for decrypting the K-encrypted data set of said selected application with said obtained encrypting key.

5 76. A system as defined in claim 75, wherein means for obtaining said encrypting key and said means for decrypting the K-encrypted data set are realized as an integrated circuit.

77. A system as defined in claim 74 or 76, wherein said integrated circuit is incorporated into said means for obtaining a user's secret key.

10

78. A system as defined in claim 73, wherein said means for deciding to use one comprises means for deciding to use a free play mode and wherein said means for playing said selected application further comprises: means, prior to said application play means, of:

15

means for receiving data from said server; and
displaying said data as an expected play time for said selected application.

20

79. A system as defined in claim 73, wherein said means for deciding to use one of said predetermined play modes comprises means for deciding to use a free play mode, and wherein said means for playing said selected application further comprises:

25

means for causing said server to obtain, as a measured play time, data of a operation period of said application play means;
means for receiving first and second data from said server; and
means for displaying, just after the completion of operation by said application play means, said first and second data as said measured play time and a total amount of play time. data as said measured play time and a total amount of play time.

30

80. A system as defined in claim 79, wherein said means for causing said server to obtain data of said operation period comprises means for informing said server of the start and the end of operation by said application play means to utilize a timer of said server.

35

81. A system as defined in claim 79, wherein said means for causing said server to obtain data of a operation period comprises:

40

means for measuring said operation period of said application play means; and
means for sending said operation period to said server for use in a calculation of said total amount of play time.

45

82. A system as defined in claim 72, wherein said means for deciding to use one comprises means for deciding to use a charged play mode and wherein said means for playing said selected application further comprises:

50

means for obtaining and sending a credit card number of said user to said server;
means responsive to a verification result of said credit card from said server for starting a next process only if said result is positive; and
means for displaying a charge for play decided based on a measured play time of said application play means and a total amount of play charges after operation of said application play means.

55

83. A system as defined in claim 82, wherein said means for playing said selected application further comprises:

60

means activated prior to operation of said application play means for displaying an expected charge and an expected total amount of charges and letting the user decide whether to play said selected application.

65

84. A system as defined in claim 82, wherein said volume control data of said distributed application package further includes a sever's public key, and wherein said means for obtaining and sending a credit card number of said user to said server comprises:

70

means for prompting said user to input said credit card number;
means for receiving a random number from said server;

means for obtaining said server's public key from said volume;

means for double-encrypting said credit card number first with said random number and then with said server's public key into a double-encrypted data;

sending said double-encrypted number to said server;

5

85. A system as defined in claim 84, wherein said means for said client obtaining and sending a credit card number of said user to said server further comprises:

means responsive to a positive result of random number check from said server for starting a next process; and
 means responsive to a negative result of said random number check from said server for displaying a message indicative of a failure in said random number check and quitting the operation for said selected application.

10

86. A system as defined in claim 72, wherein:

said means for deciding to use one comprises means for deciding to use a limit-attached play mode; and

said sending to said server includes sending a limit value associated with said mode code, and wherein said means for playing said selected application further comprises:

means operative prior to operation of said application play means for receiving from said server a limit check result indicative of whether a limit value associated with said mode code has been reached; and
 means responsive to an over limit case of said result for starting a next operation.

15

20

87. A system as defined in claim 86, wherein said limit value is one of effective date and time, allowable expiration date and time, a maximum amount of play time, and an allowable access count.

25

88. A system for controlling through a communication network a client device to play a distributed application package in one of predetermined play modes, wherein the application package contains a data set encrypted with an encrypting key (a K-encrypted data set) for each of at least one application and volume control data for use in controlling operation of the system and the client and the volume control data includes a volume ID, an issue number, an application ID for each of said applications, and a mode code for said volume or mode codes for said applications, the system comprising:

30

volume data table for storing, for each volume, said volume ID, said issue number, said mode code for said volume, and said application ID and said mode code for each of said applications;

means for receiving a service request, a volume ID, an issue number, an application ID and a mode code and other data from said client;

35

means for storing said received application ID, said received mode code and other data in appropriate fields of a record identified by said volume ID and said issue number;

means responsive to a determination that there is no record identified by said volume ID and said issue number in said volume data table for adding said record in said volume data table and storing said received application ID and mode code and said other data in relevant fields of said record; and

40

means operative on the basis of said received mode code for deciding to subsequently passing the control to means for supporting a play mode associated said received mode code.

89. A system as defined in claim 88, wherein said means for supporting a play mode at least comprises means for supporting application play means, of client, for simply playing an application identified by said received application ID, and wherein said means for supporting said application play means of said client comprises:

45

first means for associating a given volume ID with a corresponding encrypting key;

second means for associating both a given volume ID and issue number with a corresponding user's public key;

50

means for retrieving an encrypting key associated with said received volume ID from said first means;

means for retrieving a user's public key associated with said received volume ID and issue number from said second means;

means for double-encrypting said encrypting key with a pseudo random number and said user's public key into a double encrypted data; and

55

sending said double-encrypted data to said client.

90. A system as defined in claim 89, further comprising an application data table for storing data for each kind of appli-

cation, wherein said received mode code defines a free play mode, and wherein said means for supporting a play mode associated said received mode code comprises:

5 means, activated prior to an operation of said means for supporting application play means of said client, for retrieving an expected play time associated with said received application ID from said application data table; and
means for sending said expected play time to said client.

91. A system as defined in claim 89, wherein said received mode code defines a free play mode, and wherein said means for supporting a play mode associated said received mode code comprises:

10 means for measuring, as a measured play time, a duration of application play;
means for adding said measured play time to a play time meter associated with said received mode code in said volume data table to obtain a total amount of play time; and
15 means for sending said measured play time and said total amount of play time to said client.

92. A system as defined in claim 91, wherein said means for measuring a duration comprises:

20 means responsive to a notice of the start of operation by said application play means of said client for starting a timer; and
means responsive to a notice of the end of said operation for stopping said timer.

93. A system as defined in claim 91, wherein said means for measuring a duration comprises:

25 means for receiving a measured duration from said client.

94. A system as defined in claim 88, wherein said received mode code defines a charged play mode, and wherein said means for supporting a play mode associated said received mode code comprises:

30 means for receiving a credit card number of said user from said server;
means, responsive to a determination, from a verification of said credit card number, that said credit card number is not valid, for informing said client of invalidity and quitting the operation of said means for supporting a play mode;
35 means, responsive to a determination, from said verification of said credit card number, that said credit card number is valid, for informing said client of a validity and proceeding to a next operation; and
means for charging said play to said credit card number.

95. A system as defined in claim 94, wherein said means for supporting a play mode associated said received mode code further comprises:

40 means activated prior to operation of said application play means of said client for retrieving an expected charge from said application data table by using said received application ID;
means for calculating a sum of said expected charge and a value of a charge meter associated with said received volume ID or application ID depending on said received mode code;
45 means operative prior to operation of said application play means for sending said expected charge and said sum to said client; and
means responsive to a receipt of a message of quitting for quitting said means for supporting a play mode.

96. A system as defined in claim 94, wherein said means for receiving a credit card number of said user from said server comprises:

50 means for generating a pseudo random number;
means for storing said pseudo random number in memory;
means for transmitting said pseudo random number to said client;
55 means for waiting for a double-encrypted data from said client;
means for obtaining a server's secret key;
means for decrypting said double-encrypted number with said server's secret key into a decrypted random number and another decrypted data; and

means for decrypting said another encrypted data with said transmitted random number to obtain said credit card number.

97. A system as defined in claim 96, wherein said means for obtaining a user's secret key comprises means for reading said user's secret key from a portable memory of said user.

98. A system as defined in claim 96, wherein said means for receiving a credit card number of said user from said server further comprises:

means responsive to a determination, made prior to said decrypting said another, that said decrypted random number coincides with said pseudo random number which has been stored in said memory for sending an enable message to said client and proceeding to a next operation; and
means responsive to a determination, made prior to said decrypting said another, that said decrypted random number does not coincide with said pseudo random number which has been stored in said memory for sending a disable message to said client and quitting said supporting a play mode.

99. A system as defined in claim 88, wherein:

said received mode code defines a limit-attached play mode; and
means for receiving a service request further receives a limit value associated with said mode code, and wherein said means for supporting a play mode associated said received mode code comprises:
means for proceeding to a next operation only if the value of a software meter associated with said mode code in said volume data table is under said limit value; and
means for sending a message informing an over limit to said client and quitting the operation of said means for supporting a play mode associated said received mode code if the value of a software meter associated with said mode code in said volume data table is not under said limit value.

100.A system as defined in claim 99, wherein said limit value is one of effective date and time, allowable expiration date and time, a maximum amount of play time, and an allowable access count.

101.A system as defined in any of claims 54, 73 and 75, wherein said means for obtaining a user's secret key comprises means for reading said user's secret key from a portable memory of said user.

102.A system as defined in claim 28 or 29, wherein said means for obtaining said secret key comprises means for reading said user's secret key from a portable memory of said user.

103.A method as defined in any of claims 10, 11, 19, 21, 22 and 55, wherein said application package is recorded on a package media.

104.A method as defined in claim 103, wherein said package media is of a write-once type, and said client is a system capable of playing said package media of said write-once type.

105.An application package as defined in claim 1, wherein said package media is distributed to a purchaser thereof or a subscriber thereof via a transmission media.

106.A system as defined in any of claims 28, 29, 37, 39, 40, 70 and 88, wherein said application package is recorded on a package media.

107.A system as defined in claim 106, wherein said application package is recorded on a package media of a write-once type.

108.A system as defined in claim 106, wherein at least a part of said volume control data is recorded, after manufacturing said package media, in an area different from a data area where said at least one application is recorded.

109.A system as defined in claim 108, wherein said client is a system provided with means for playing said package media of said write-once type.

110.A system as defined in any of claims 28, 29, 37, 39, 40, 70 and 88, wherein said application package is recorded

on a DVD and at least a part of said volume control data is recorded, after manufacturing said package media, in a BCA (burst cutting area) of the DVD, and wherein said client is a system provided with means for playing said DVD.

5 111. A method as defined in any of claims 10, 11, 19, 21, 22, 43 and 55, wherein the application package has been distributed to a purchaser thereof or a subscriber via a transmission media and at least a part of said volume control data has been added to said application package after preparing said application package.

10 112. A system as defined in any of claims 28, 29, 37, 39, 40, 49, 70 and 88, wherein said application package has been distributed to a purchaser thereof or a subscriber thereof via a transmission media and at least a part of said volume control data has been added to said application package after preparing said application package.

15

20

25

30

35

40

45

50

55

FIG. 1

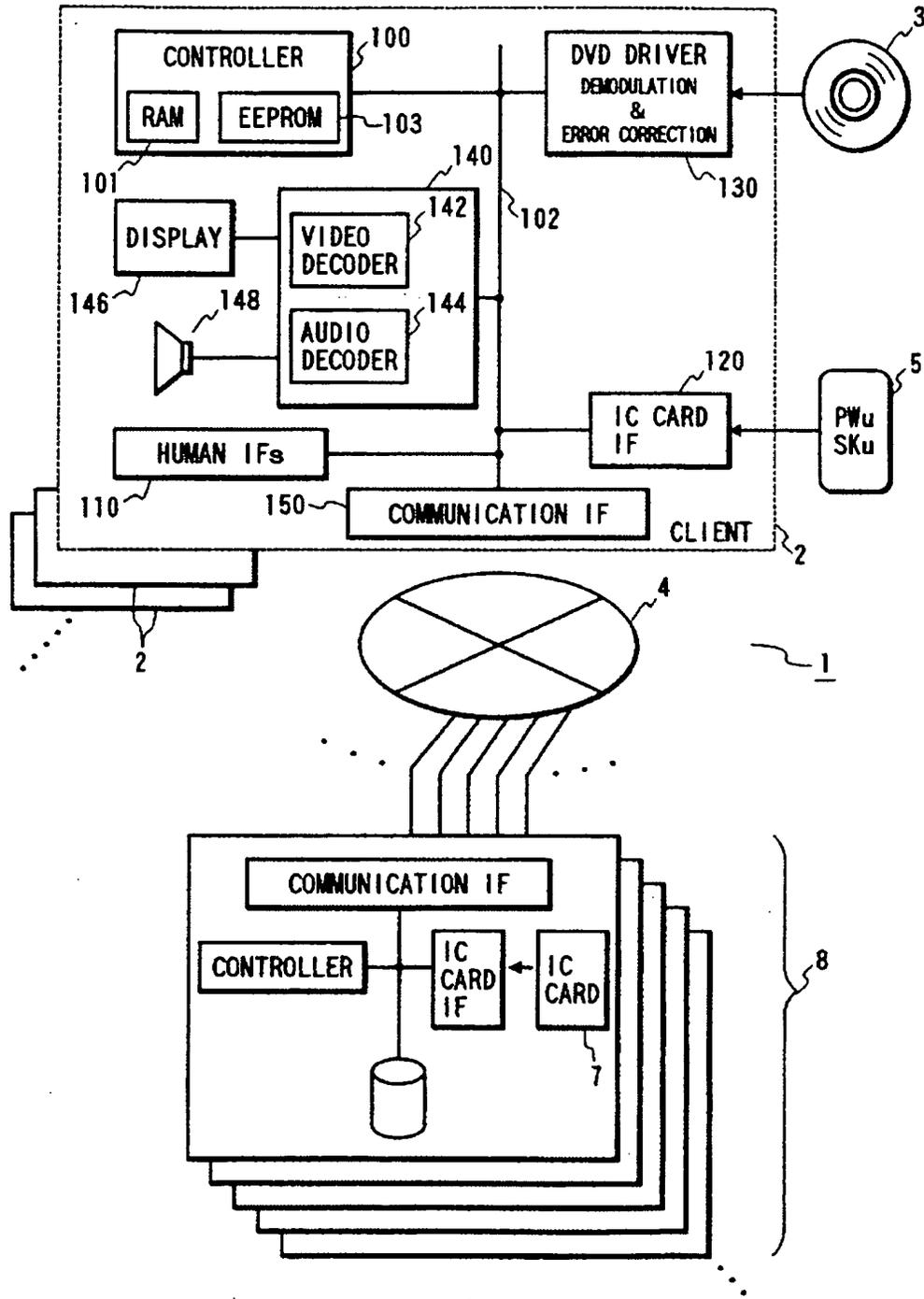


FIG. 2

20

BURST CUTTING AREA	DISTRIBUTION DESCRIPTOR	23
DATA AREA	VOLUME DESCRIPTOR	22
	VOLUME CONTROL PROGRAM	24
	APPLICATION (APPLICATION) : :	21

FIG. 3

VOLUME IDENTIFIER (VID _v)	25
PROVIDER IDENTIFIER (PID _p)	26
: :	
VOLUME CREATION DATE AND TIME	27
VOLUME EFFECTIVE DATA AND TIME	28
: :	
(APPLICATION IDENTIFIER 1)	29
(APPLICATION IDENTIFIER 2)	
: :	
: :	

FIG. 4

23

VOLUME ISSUE NUMBER (NO _v i)	30
⋮	
SERVER PUBLIC KEY (PK1)	31
(PK2)	
⋮	
PK _u -ENCRYPTED AP-ENCRYPTING KEY (K _v) = e ₁ (PK _u , K _v)	32
⋮	
SALES DATE AND TIME	33
TERMS-OF-USE CODE PLUS LIMIT VALUE FOR THE VOLUME	34
(TERMS-OF-USE CODE PLUS LIMIT VALUE FOR APPLICATION 1)	35
(TERMS-OF-USE CODE PLUS LIMIT VALUE FOR APPLICATION 2)	
⋮	

FIG. 5

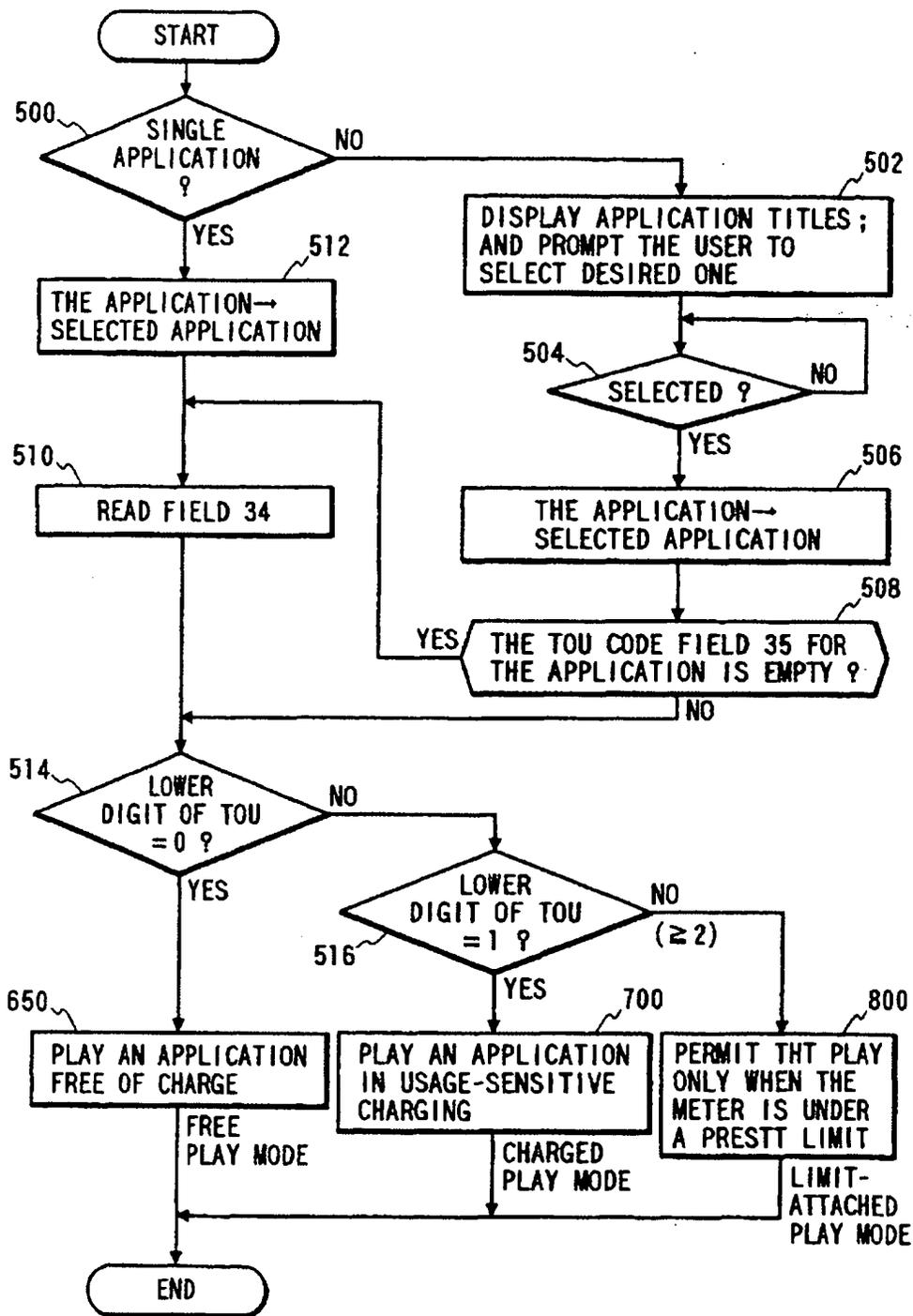


FIG. 6A

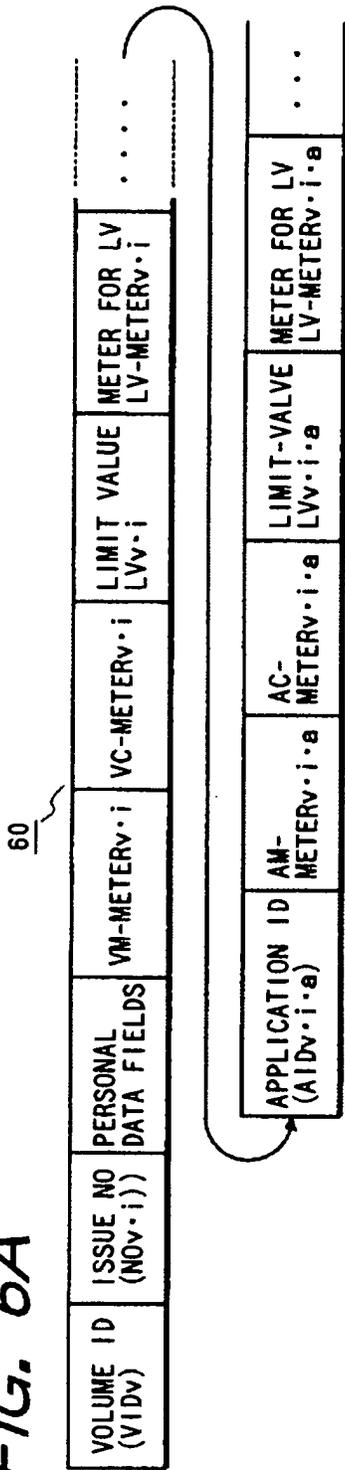


FIG. 6B

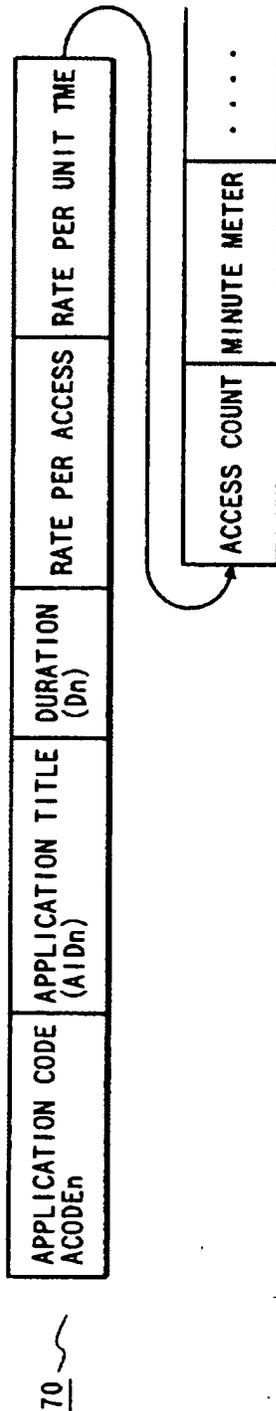


FIG. 7



FIG. 8A

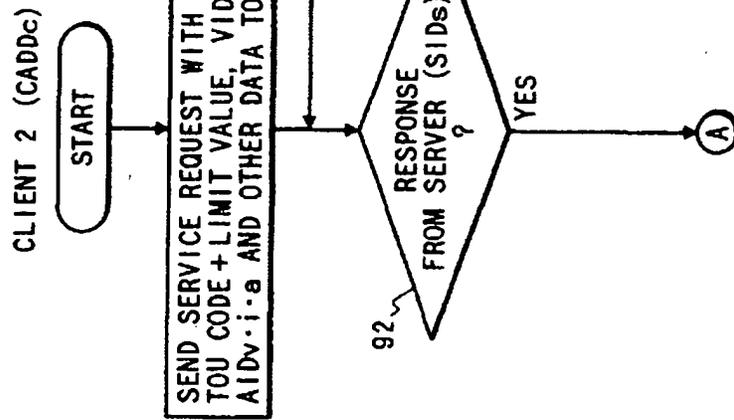
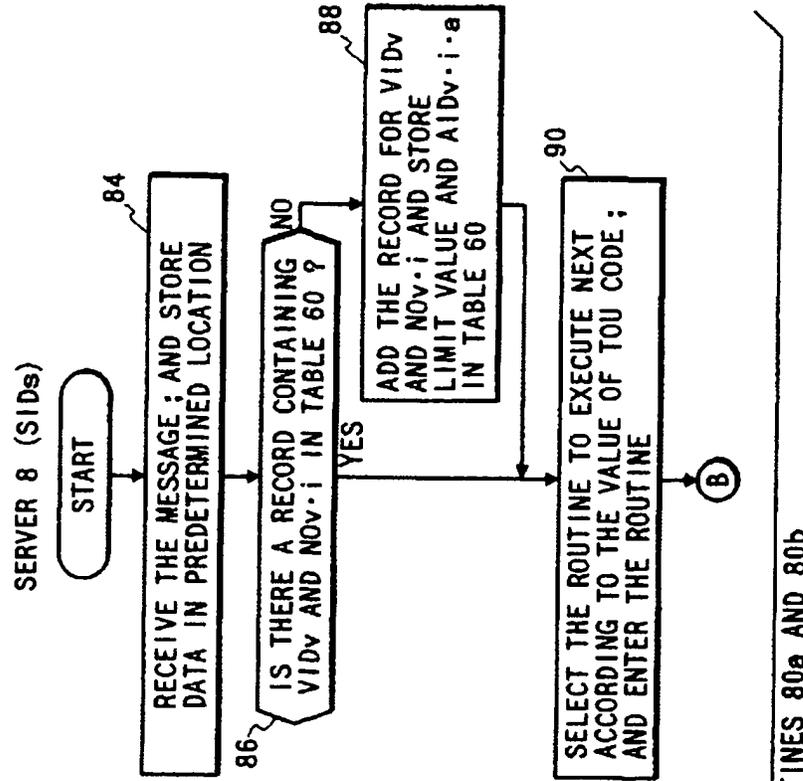


FIG. 8B



INITIAL ROUTINES 80a AND 80b

FIG. 9

PLAY AN APPLICATION FREE OF CHARGE

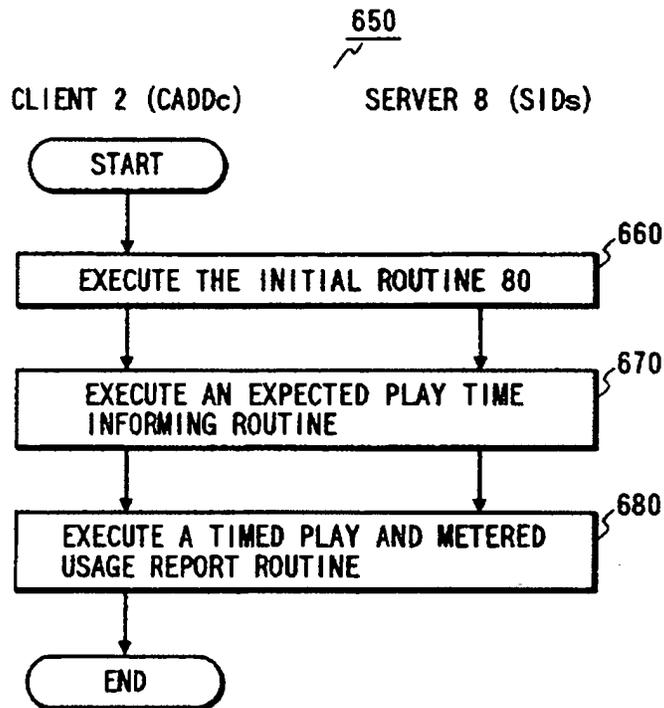


FIG. 10A

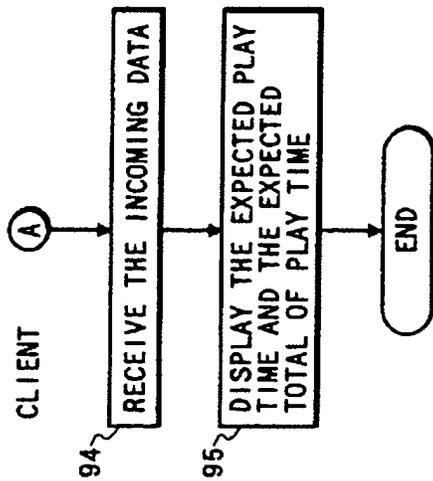
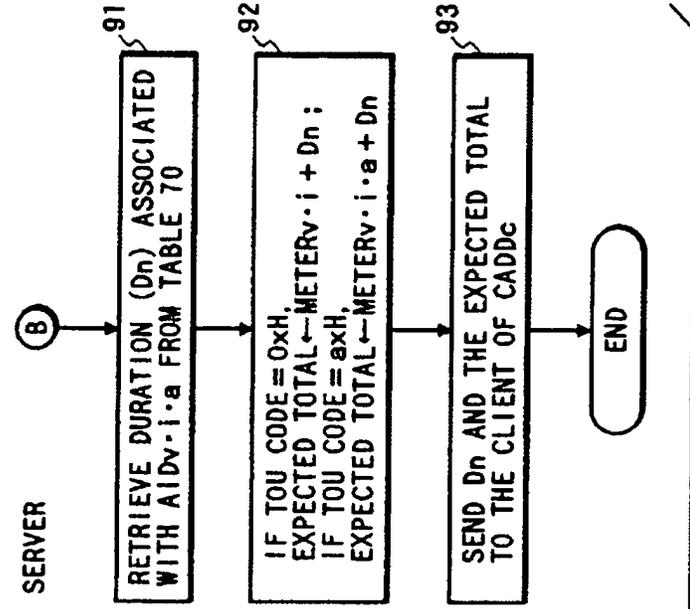


FIG. 10B



EXPECTED PLAY TIME INFORMING ROUTINES 97a AND 97b

FIG. 11A

TIMED PLAY AND METERED USAGE REPORT ROUTINES 675a AND 675b

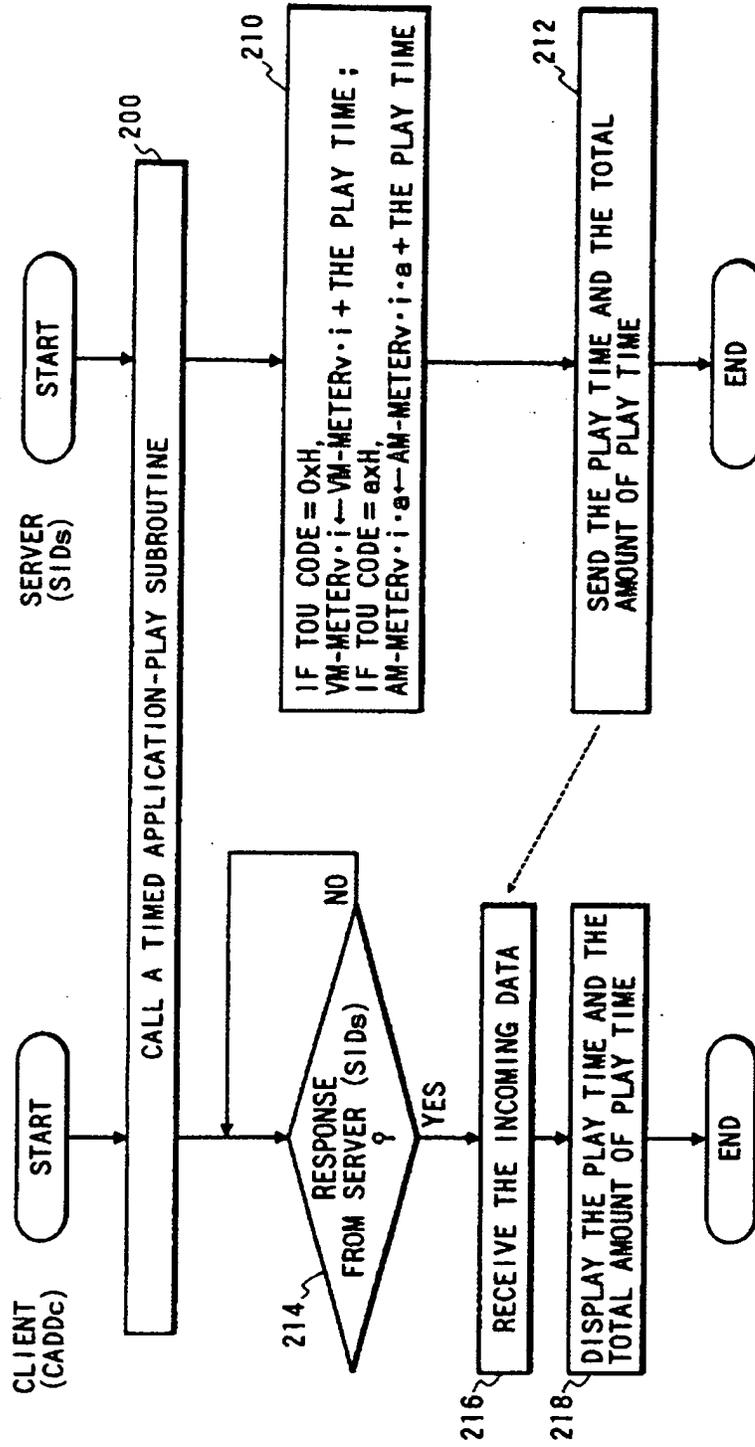


FIG. 12A

FIG. 12B

TIMED APPLICATION-PLAY SUBROUTINES

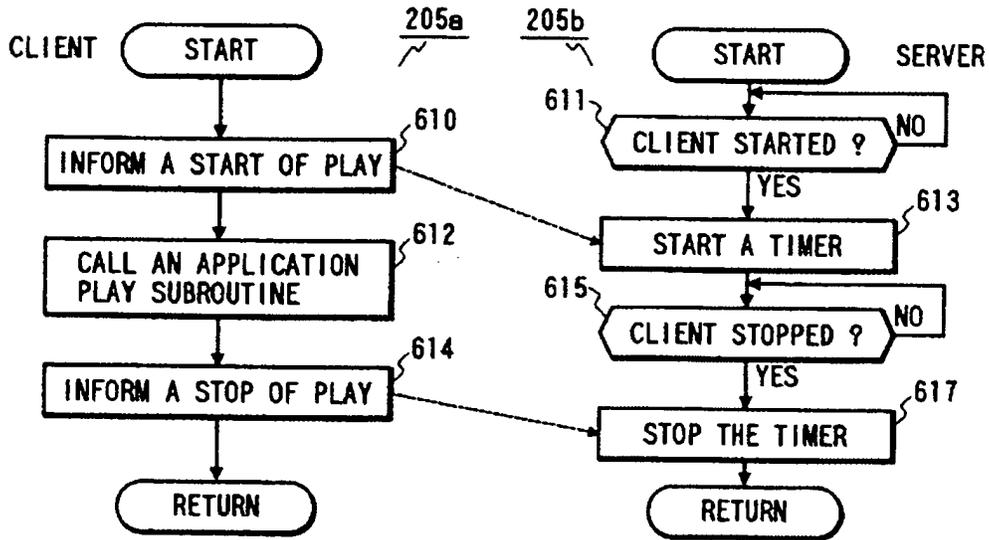


FIG. 13A

FIG. 13B

TIMED APPLICATION-PLAY SUBROUTINES

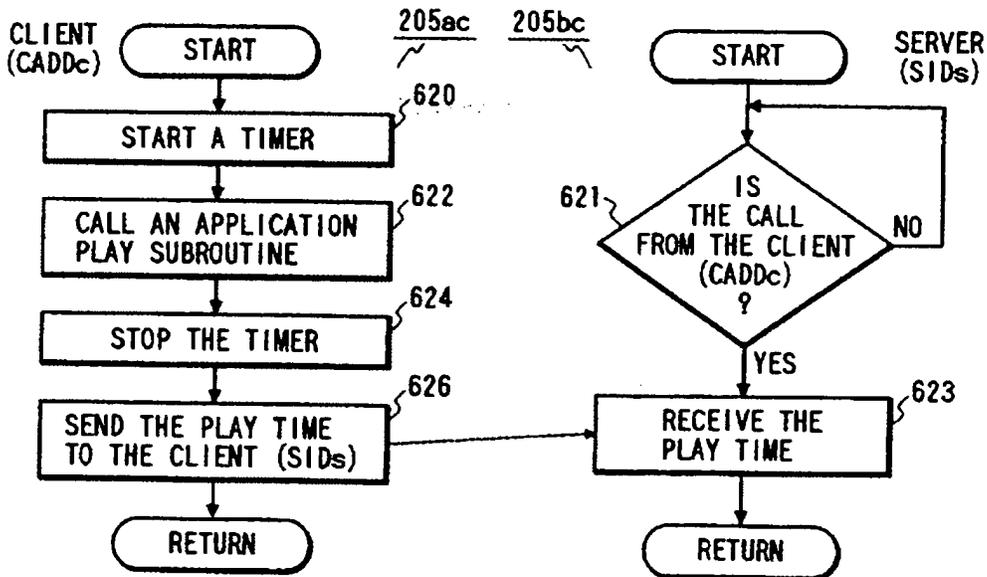


FIG. 14

APPLICATION PLAY SUBROUTINE

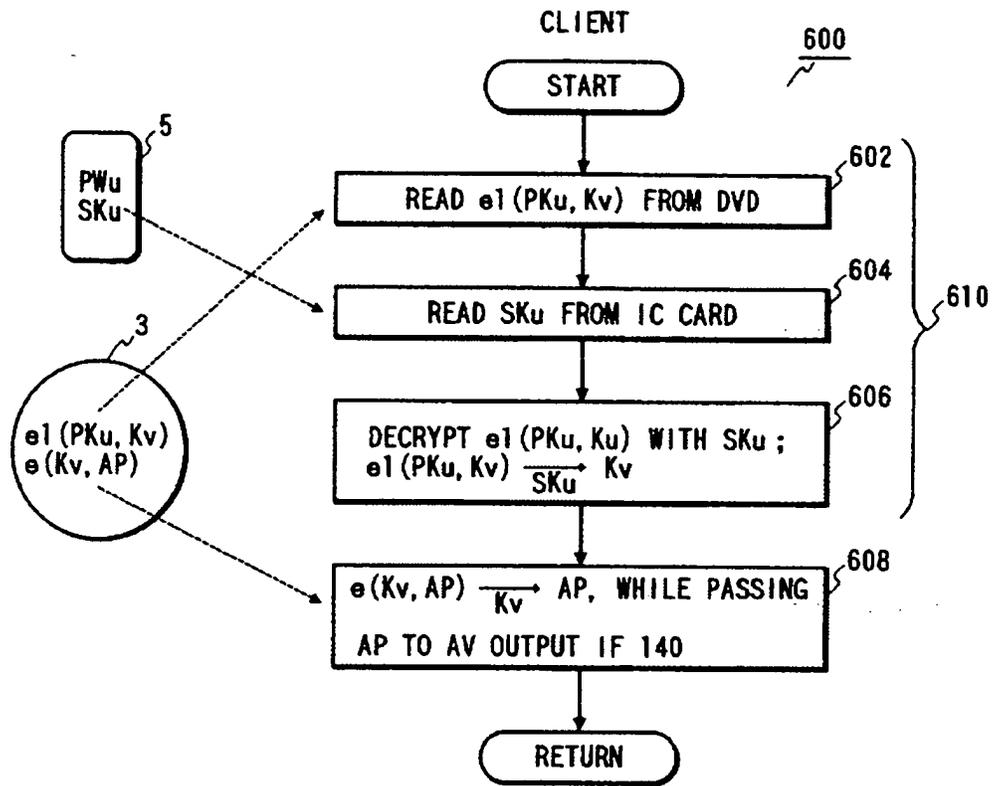


FIG. 15

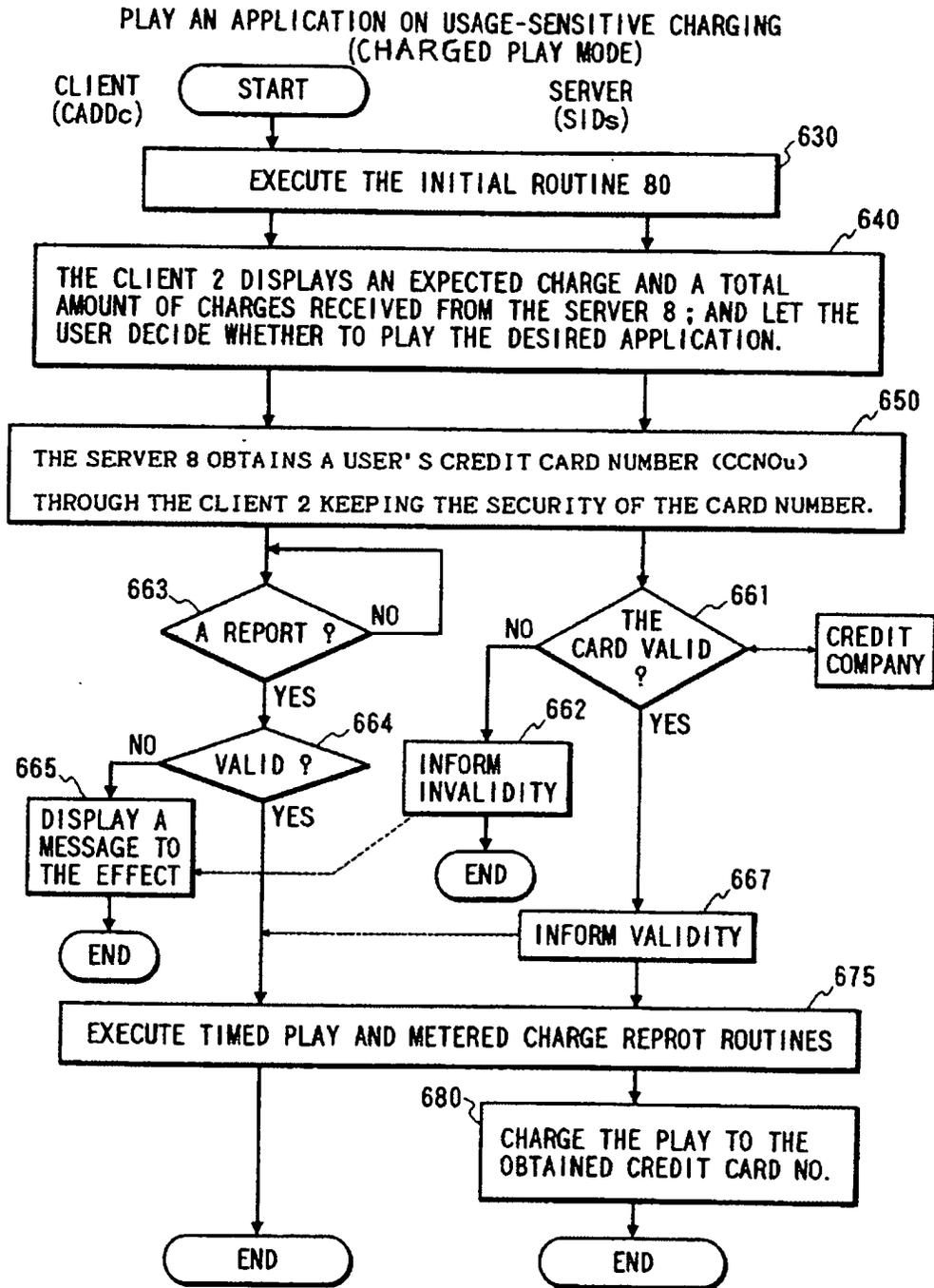


FIG. 16A

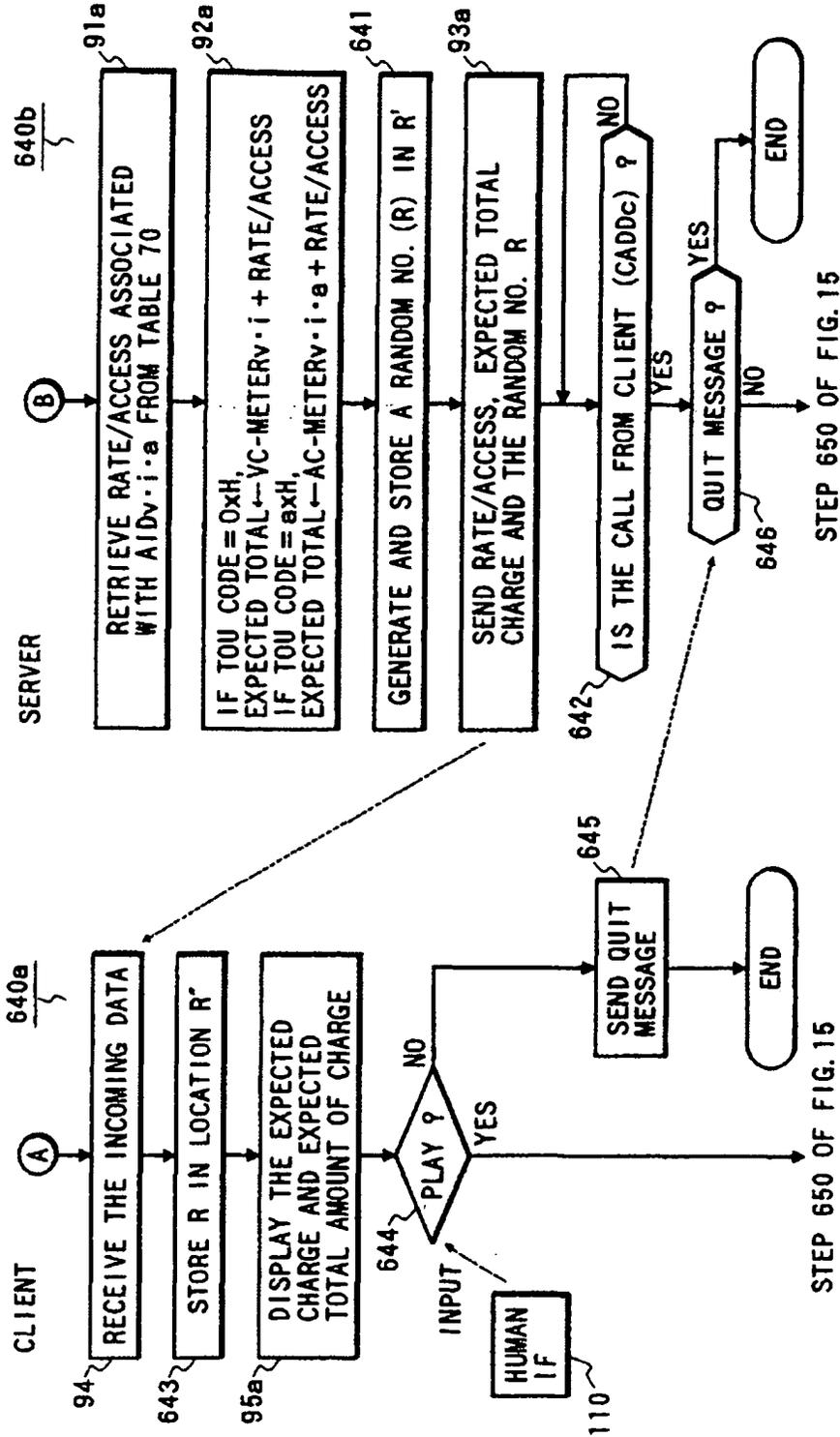


FIG. 16B

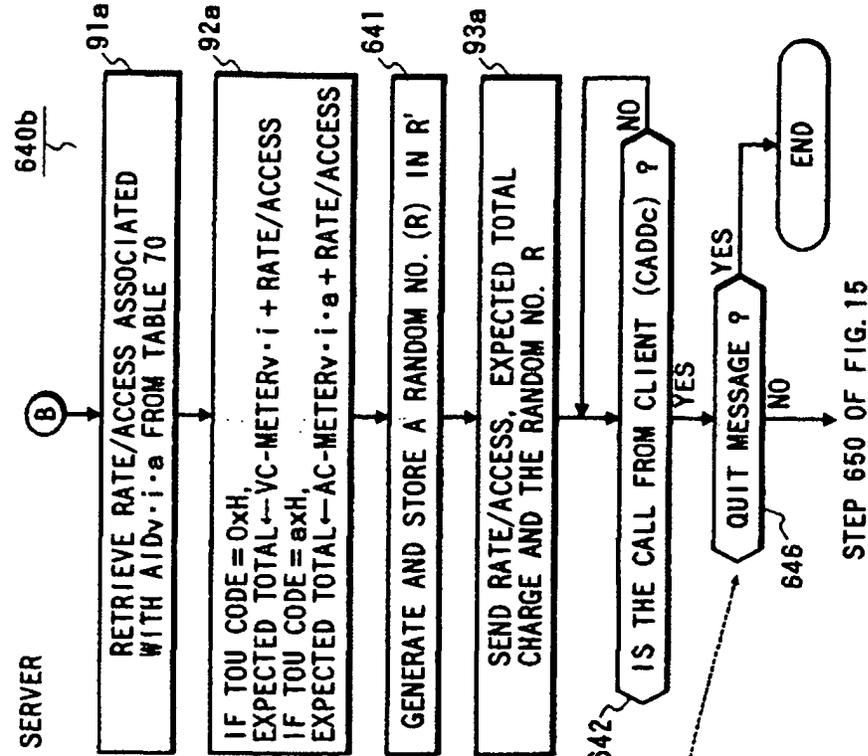


FIG. 17A

FROM BLOCK 640 OF FIG. 15
(STEP 644 OF FIG. 16A)

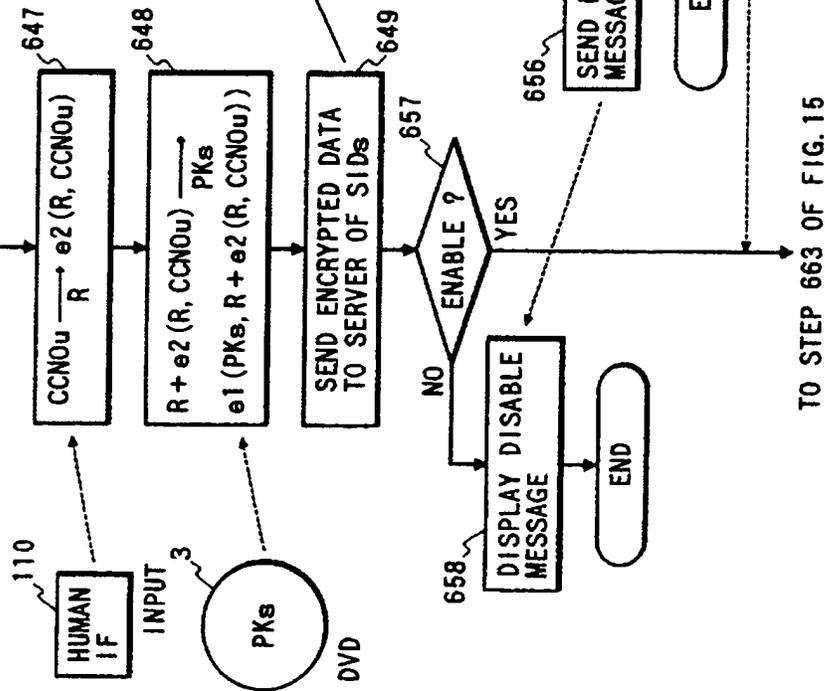


FIG. 17B

FROM BLOCK 640 OF FIG. 15
(STEP 646 OF FIG. 16B)

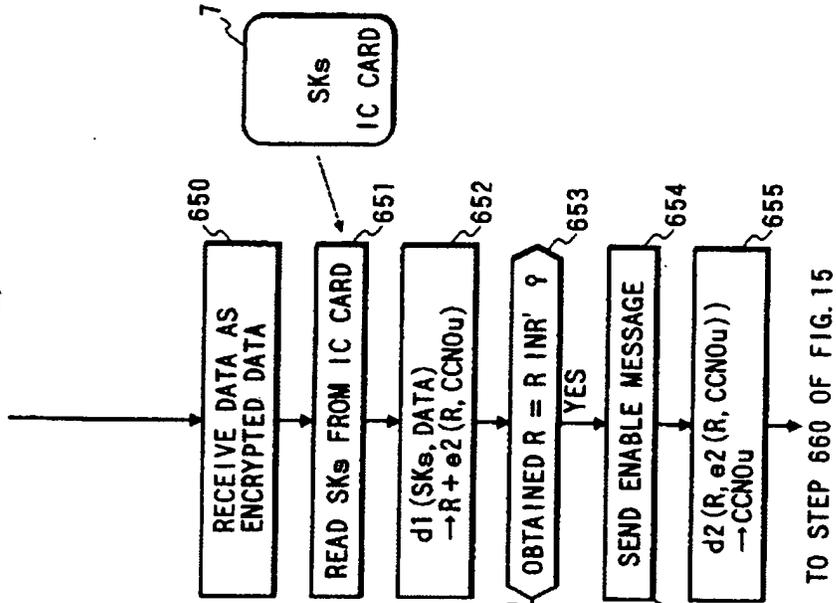


FIG. 18A
 FIG. 18B

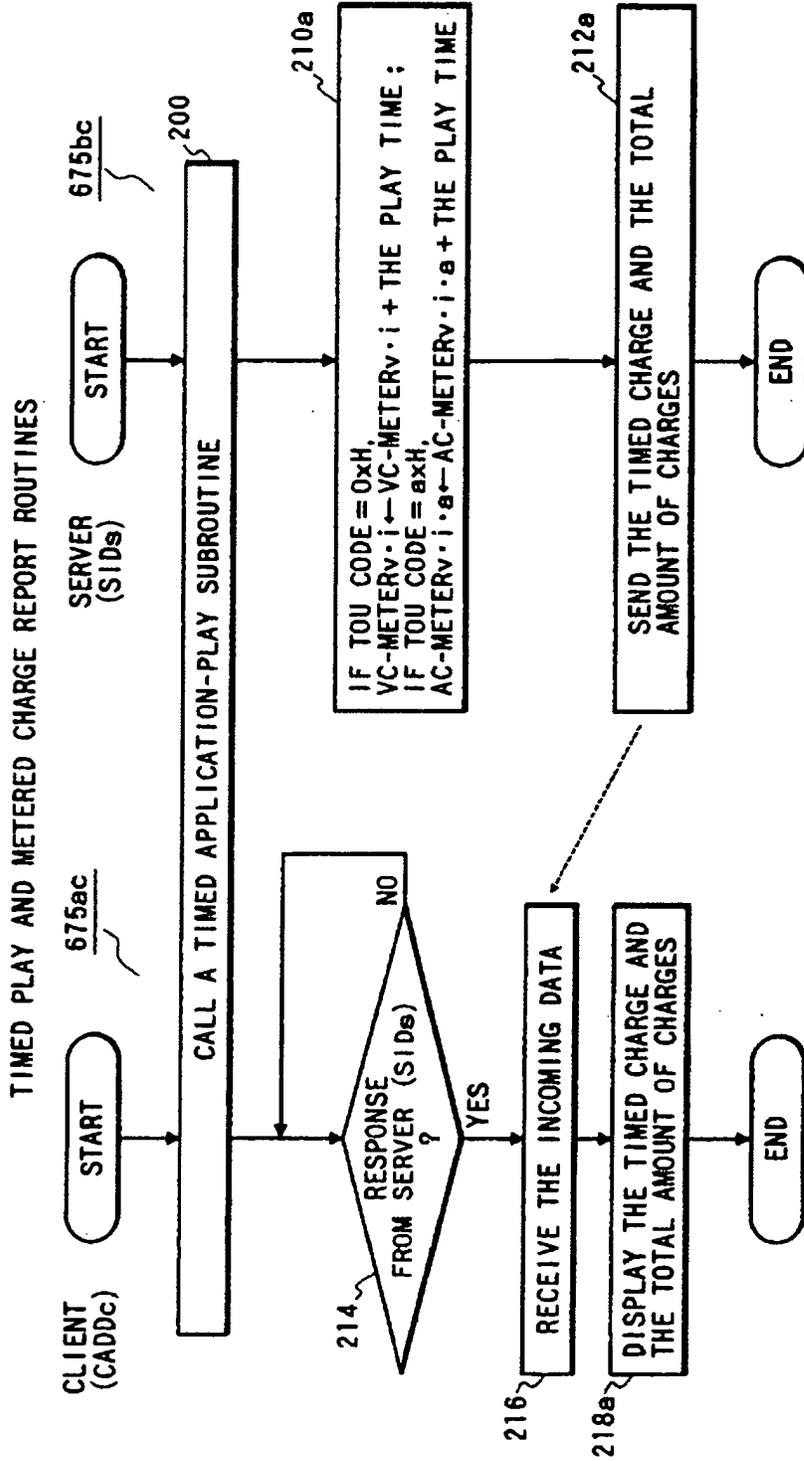


FIG. 19

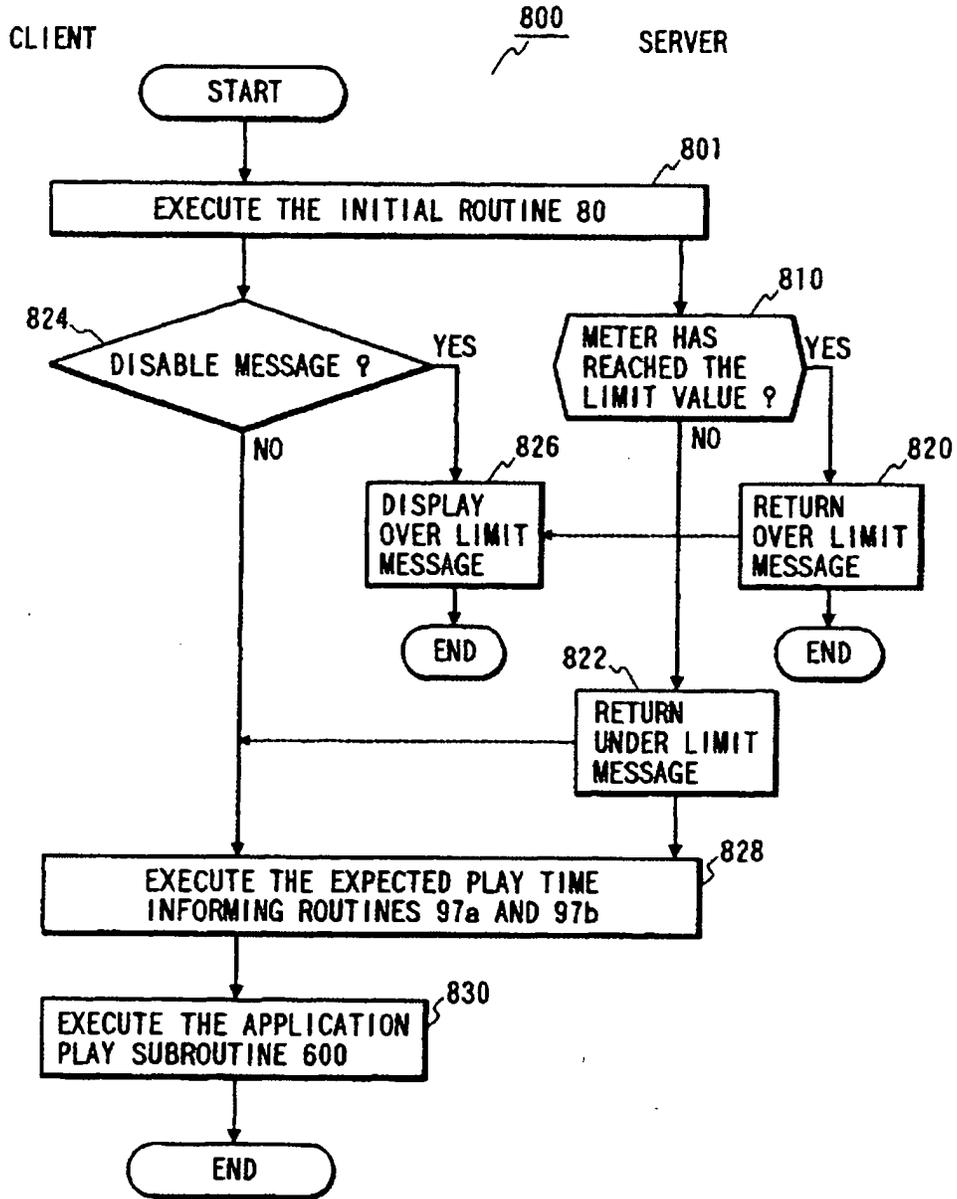


FIG. 20A

VIDv	Kv
VID1	K1
VID2	K2
⋮	⋮

FIG. 20B

VIDv	NOv·i	PKu
VID1	NO1·1	PK347020
	NO1·2	PK001031
	⋮	⋮
VID2	NO1·365	PK314162
	NO2·1	PK141421
⋮	⋮	⋮
VID3	NO2·77	PK789012
	NO3·1	PK123456
⋮	⋮	⋮

FIG. 20C

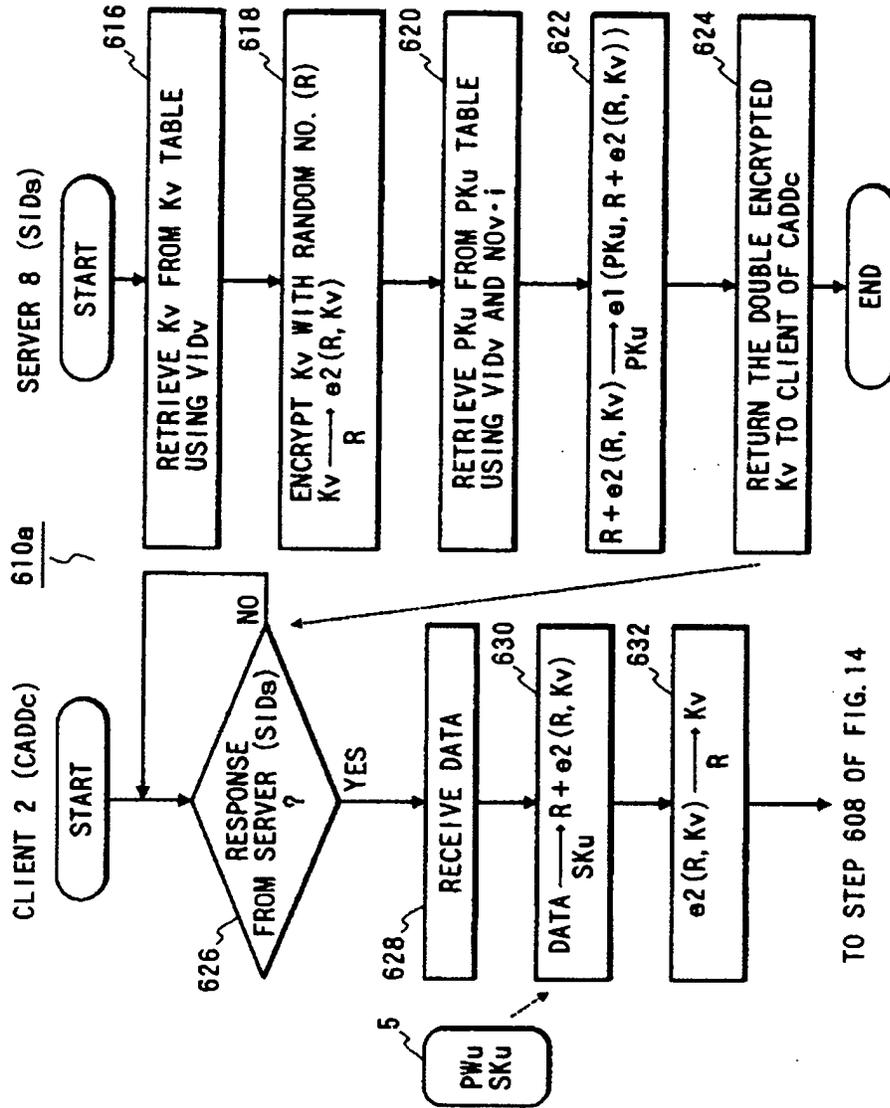


FIG. 21

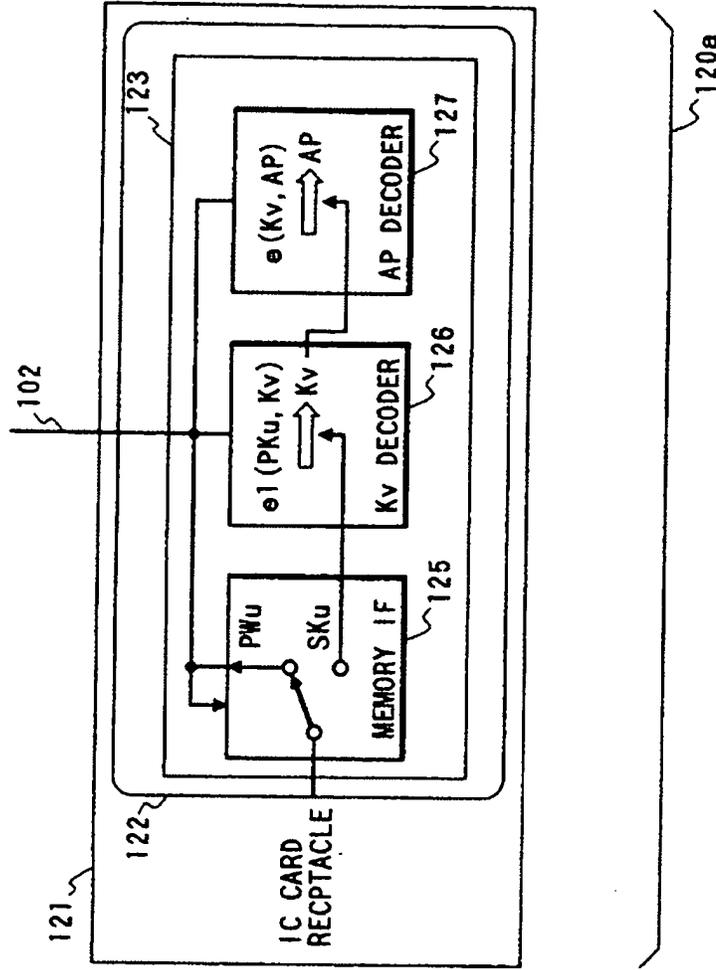


FIG. 22

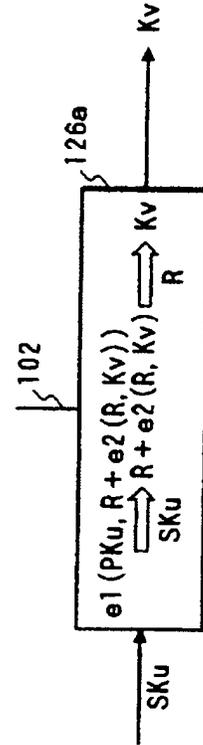


FIG. 23

THE HIGHER DIGIT OF TERMS-OF-USE CODE (HEXADECIMAL)	THE TERMS-OF-USE CODE IS APPLIED TO :
0	THE ENTIRE VOLUME
1	APPLICATION 1
2	APPLICATION 2
⋮	⋮

XYH (X, Y = 1, 2, ..., F)

THE LOWER DIGIT OF TERMS-OF-USE CODE (HEXADECIMAL)	CORRESPONDING LIMIT VALUE
0	NONE
1	NONE
2	THE EFFECTIVE DATE AND TIME
3	THE ALLOWABLE EXPIRATION DATE AND TIME
4	THE MAXIMUM AMOUNT OF USED PERIOD
5	THE ALLOWABLE ACCESS COUNT
⋮	⋮

FIG. 24

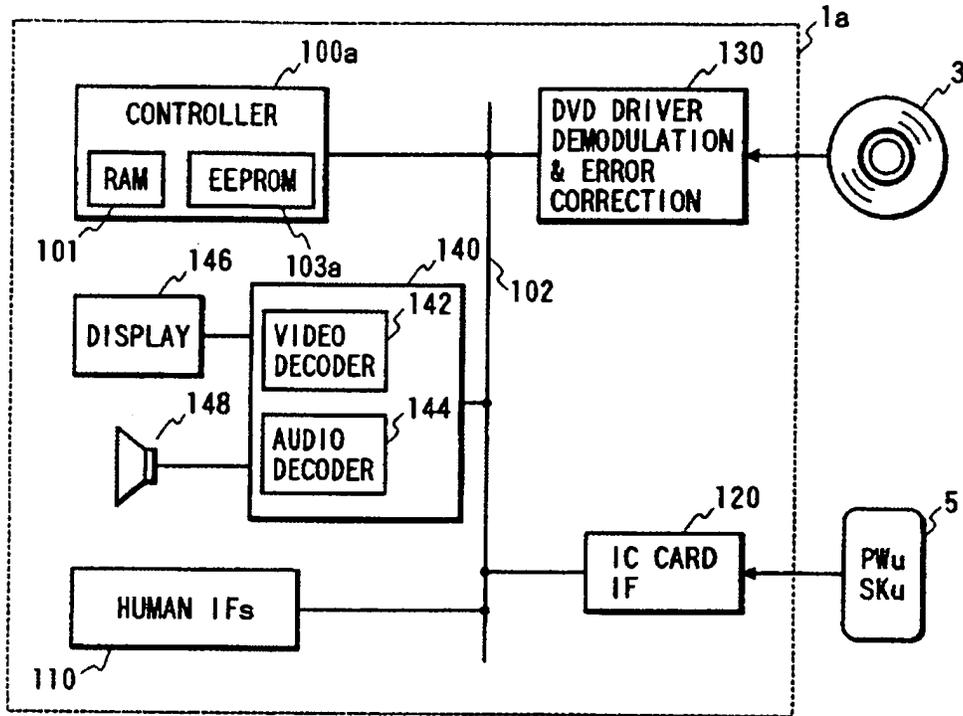


FIG. 26

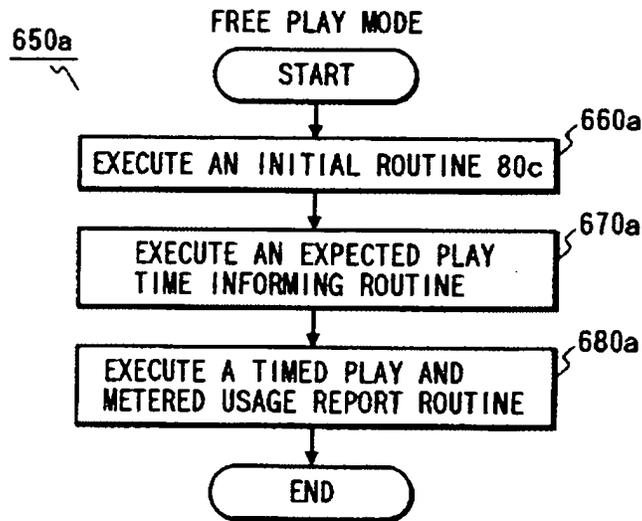


FIG. 25

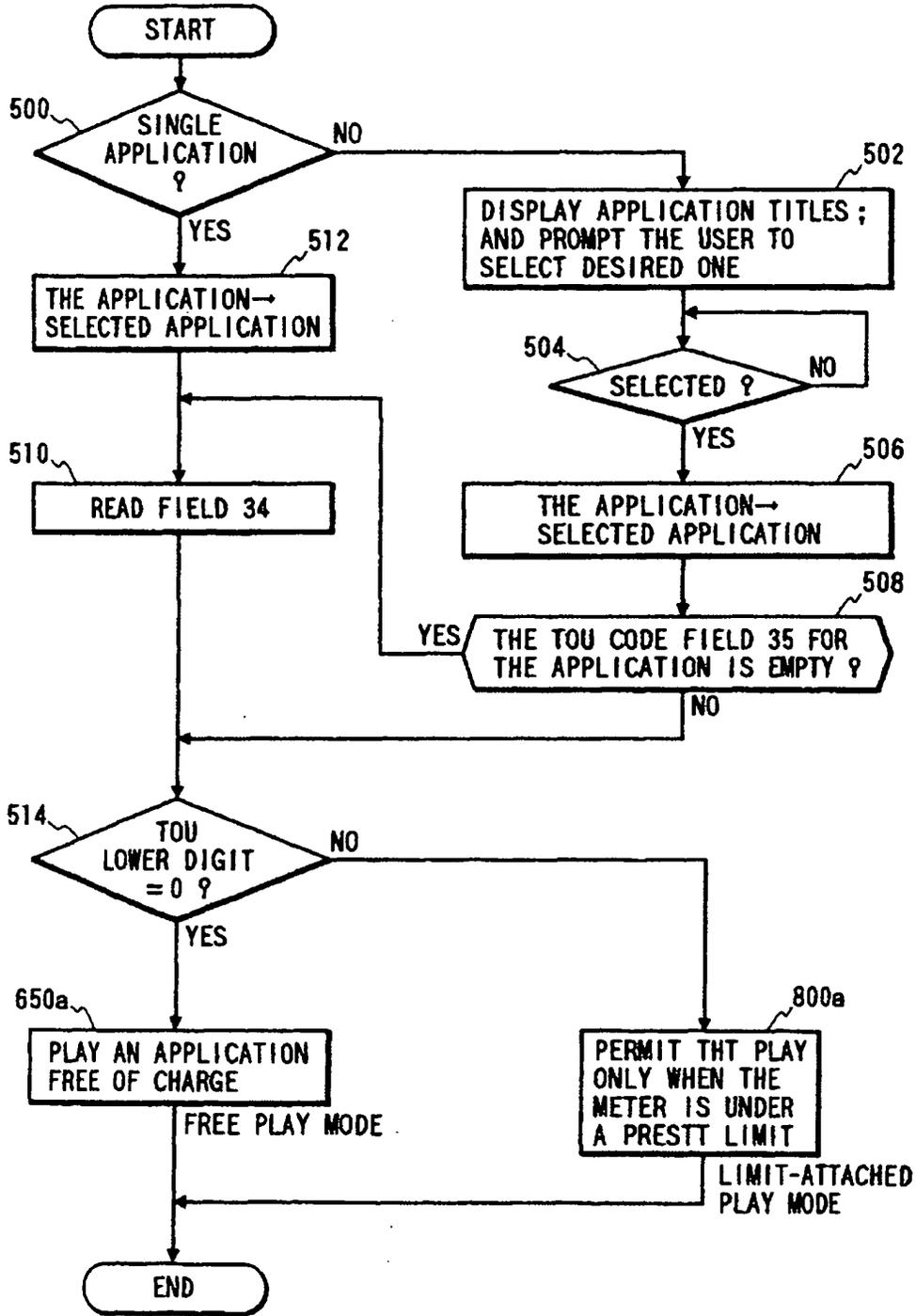


FIG. 27

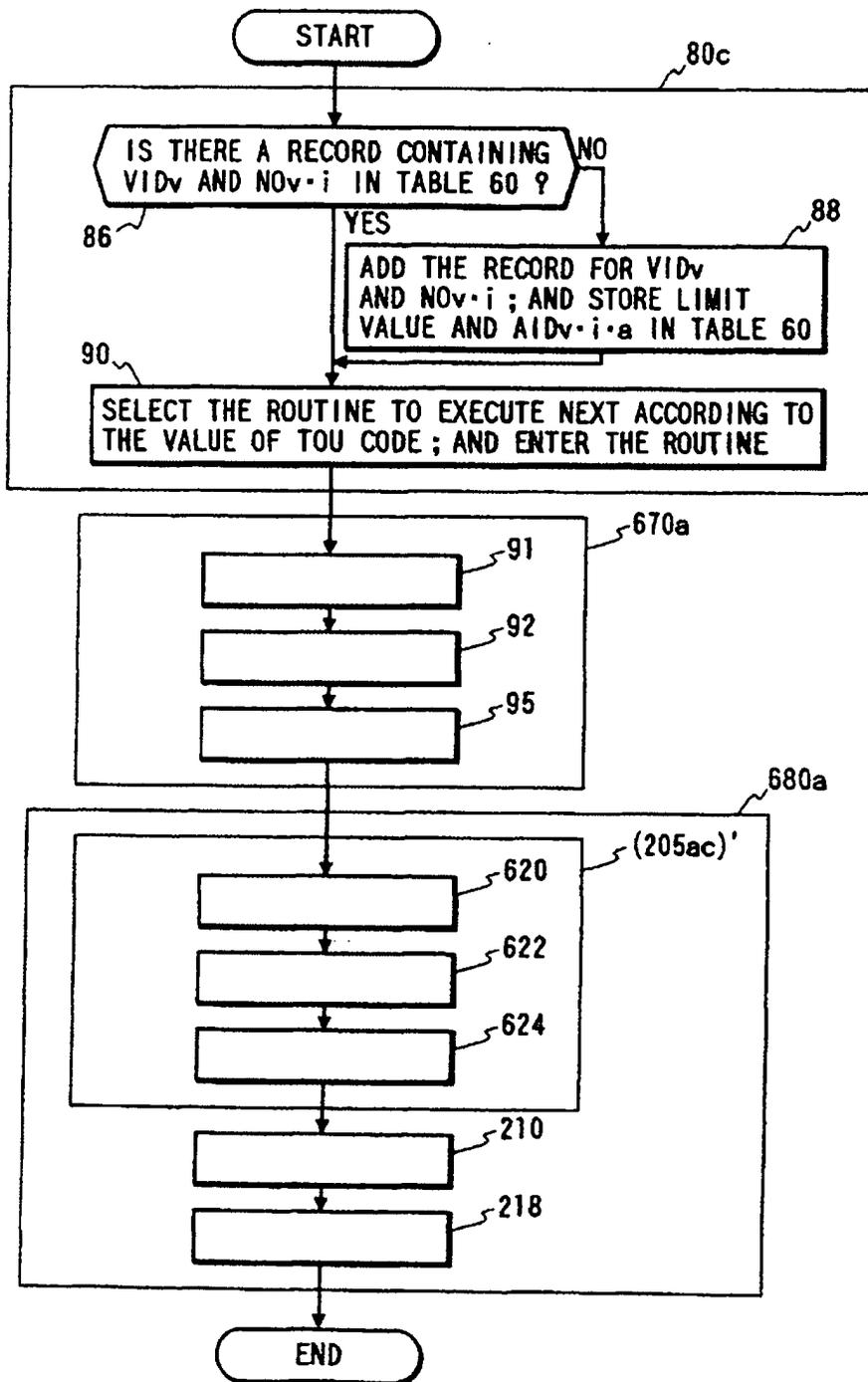
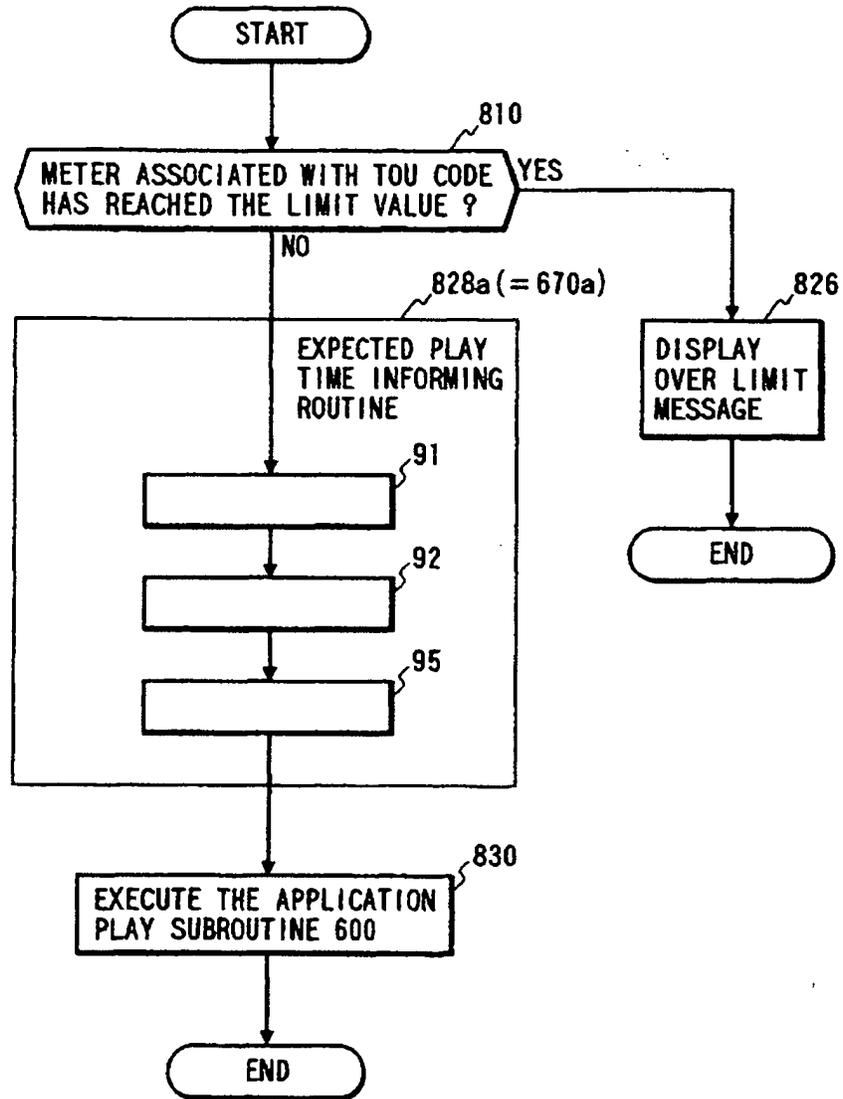


FIG. 28

LIMIT-ATTACHED PLAY MODE





Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) EP 0 892 521 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
20.01.1999 Bulletin 1999/03

(51) Int Cl.⁶: H04L 9/32

(21) Application number: 98305646.6

(22) Date of filing: 15.07.1998

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(72) Inventor: Zamek, Steven
Palo Alto, California 94303 (US)

(74) Representative: Jehan, Robert et al
Williams, Powell & Associates,
4 St Paul's Churchyard
London EC4M 8AY (GB)

(30) Priority: 15.07.1997 US 892792

(71) Applicant: Hewlett-Packard Company
Palo Alto, California 94304 (US)

(54) Method and apparatus for long term verification of digital signatures

(57) The time over which a digital signature can be verified is extended well beyond the expiration of any or all of the certificates upon which that signature depends. In a "save state" approach, an archive facility is used to store public key infrastructure (PKI) state, e.g. cryptographic information, such as certificates and certificate revocation lists (CRLs), in addition to non-cryptographic information, such as trust policy statements or the document itself. This information comprises all that is necessary to re-create the signature verification process at a later time. When a user wants to verify the signature on a document, possibly years later, a long term signature verification (LTSV) server re-creates the precise

state of the PKI at the time the document was originally submitted. The LTSV server restores the state, and the signature verification process executes the exact process it performed (or would have performed) years earlier. Another embodiment of the invention combines the strength of cryptography with the proven resilience of (non-public key) technology and procedures currently associated with secure data stores by saving the PKI state for future verification; and protecting the PKI state information from intrusion by maintaining it in a secure storage facility which is protected by services, such as firewalls, access control mechanisms, audit facilities, intrusion detection facilities, physical isolation, and network isolation.

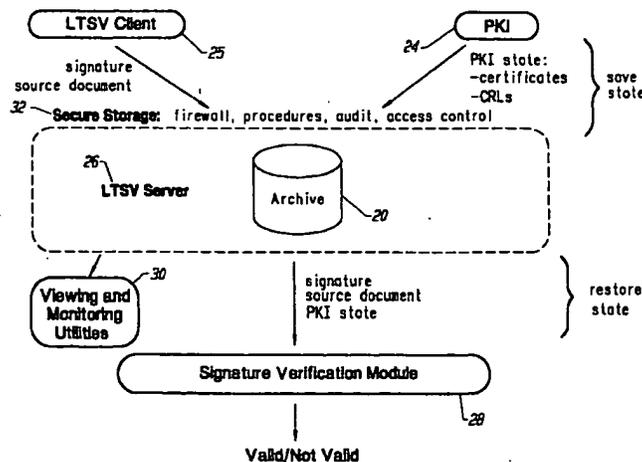


FIG. 3

EP 0 892 521 A2

Description

This invention relates to a method and apparatus for the long term verification of digital signatures.

The technology of digital signatures opens up the likelihood of increased use of digital networks (including the Internet) for electronic commerce. It is now feasible to send and receive digitally signed documents that represent transactions of some value to one or more parties.

Currently, a digital signature is verifiable only as long as the digital certificates upon which it depends have not expired. Given the expectation that a certificate's life span is in the area of one to two years duration, current technology does not support the emerging needs of the electronic commerce market, where the durability of digital signatures over time is a requirement.

For certain applications, the recipient of digitally signed documents should be able to verify the authenticity of a document years after the document was signed, just as the document's authenticity can be verified at the time of signing. Unfortunately, the current state of the technology does not provide for the verification of these digital signatures after certificate expiration because it is the nature of keys and certificates used for signing and encrypting documents to expire after a specific period of time (typically after a year or two). This is due, at least in part, to the fact that the strength of keys is expected to degrade over time because of such factors as improvements in computing speed and breakthroughs in cryptanalysis. Moreover, the longer the key is in use, the longer that an adversary has to attempt to crack the key. Therefore, it is standard practice to replace keys periodically. This is why certificates have specific expiration dates.

An examination of the current state of the technology reveals that a digital signature verification module would fail if presented with a request to verify a signed document in which any of the associated certificates had expired. Fig. 1 is a block schematic diagram illustrating certification expiration. This simple example demonstrates that, given a certificate 10 having a two-year life span (e.g. from 4/1/96 to 4/1/98), a signature could be successfully verified six months (e.g. on 10/1/96) after certificate issuance (100); but this same signature would not be successfully verified three years later (e.g. on 4/1/99) (102). This behavior is clearly unacceptable if the duration of a document, for example contract, must extend beyond the duration of the certificates' life.

Further, some current systems use certificate revocation lists (CRLs) to revoke certificates and remove them therefrom, once those certificates expire. This means that a record of those CRLs generally disappears, making long term signature verification impossible using known techniques.

It is known to reconstruct past trust (see A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, pp. 583 (1996)). In this ap-

proach, both signature reverification relative to a past point in time and resolution of disputes may require reconstruction of chains of trust from a past point in time. This requires archival of keying material and related information for reconstruction of past chains of trust. Direct reconstruction of such past chains is taught to be unnecessary if a notarizing agent is used. A notarizing agent is defined as a general service capable not only of ascertaining the existence of a document at a certain time, but of vouching for the truth of more general statements at certain points in time. The original verification of the notary is taught to establish the existence of a trust chain at that point in time, and subsequently its record thereof is taught to serve as proof of prior validity. It is taught that details of the original trust chain may be recorded for audit purposes. It is not taught that a document can be verified based upon the existence of expired certificates. Rather, reliance is placed upon the use of the notarizing agent. It is further taught that the archived keying material can be used as evidence at a future time to allow resolution of disputed signatures by non-automated procedures.

It would be advantageous to provide a technique for extending the time over which the authenticity and integrity of digital signatures can be accurately verified beyond the time that any relevant certificates expire.

The present invention seeks to provide improved signature verification.

According to an aspect of the present invention there is provided a method of enabling long term verification of digital signatures as specified in claim 1.

According to another aspect of the present invention there is provided apparatus as specified in claim 11.

The preferred embodiment provides a method and apparatus which effectively extends the time over which a digital signature can be verified, *i.e.* well beyond the expiration of any or all of the certificates upon which that signature depends. The invention can be used for any application domain where users want digital signatures to be applied to long lasting documents (e.g. contracts), and be independently verifiable years or decades after signing the document. The preferred embodiment provides two alternative approaches to constructing a solution which delivers long term signature verification (LTSV).

One embodiment of the invention provides an approach for solving the LTSV problem that is referred to herein as the "save state" approach. This embodiment of the invention largely entails the use of cryptographic information and techniques. Thus, an archive facility is used to store the public key infrastructure (PKI) state, e.g. cryptographic information, such as certificates and CRLs, in addition to the document itself. This information comprises all that is necessary to re-create the signature verification process at a later time. It may also be desirable to store the source document separately from the cryptographic information (such as the signature, certificates, and CRLs) for reasons of privacy. For ex-

ample, a user may want to have control over the source document. The PKI state information may contain either or both of cryptographically protected information, such as certificates and CRLs, and information that is not cryptographically protected, such as the public key of a root certification authority or policy information.

When a user wants to reverify the signature on a document, possibly years later, an LTSV server re-creates the precise state of the PKI at the time the document was originally submitted. The LTSV server restores the state, and the signature verification process executes the exact process it performed (or would have performed) years earlier. The time used as the basis for re-creation of the signature verification process does not have to be the time of submittal. Rather, the time could be some other relevant time, such as when a document was signed by the originator or when it was verified by a recipient.

Another embodiment of the invention combines the strength of cryptography with the proven resilience of (non-public key) technology and procedures currently associated with secure data stores. An example of this embodiment provides a mechanism that:

- Saves the PKI state for future reverification; and
- Protects the PKI state information from intrusion by either maintaining it in a secure storage facility which is protected by services, such as firewalls, access control mechanisms, audit facilities, intrusion detection facilities, physical isolation, and network isolation; and/or employing a cryptographic protection mechanism, for example using the LTSV server to sign the PKI state information or using a keyed hash algorithm.

In addition, other non-cryptographic features may be added to such approaches to deliver a highly secure and trusted LTSV solution, including, for example utilities for viewing the PKI state information (cryptographic as well as non-cryptographic) and visually monitoring the security of the system. These utilities can be used to provide visual evidence for purposes of dispute resolution.

One enhancement to the secure storage approach herein disclosed maintains certain evidence, such as certificate chains, in an archive. This information need not be used for actual reverification, but merely as supporting evidence in case of a dispute.

An embodiment of the present invention is described below, by way of example only, with reference to the accompanying drawings, in which:

Fig. 1 is a block schematic diagram illustrating certification expiration;

Fig. 2 is a block schematic diagram illustrating a "save state" embodiment of the invention;

Fig. 3 is a block schematic diagram illustrating a "save state" "secure storage" embodiment of the invention;

Fig. 4 is a flow diagram that provides two alternative scenarios that illustrate the applicability of time stamps to the preferred embodiments;

Figs. 5a-5c provide block schematic diagrams that illustrate three long term signature verification usage scenarios;

Fig. 6 is a block schematic diagram that illustrates trust between two entities ; and

Fig. 7 is a block schematic diagram that illustrates a long term signature verification trust model.

The meanings of some of the terms used herein may differ somewhat from common usage. The following definitions are meant to clarify the meaning of each in the context of its usage herein.

Archive: Any facility for the storage and retrieval of electronic information.

Certificate: An artifact upon which digital signatures are based. A certificate securely binds an entity with that entity's public key.

Cryptographic Refresh: A means of solving the key degradation problem when storing cryptographic information for long periods of time. The process involves re-encoding the old cryptographic artifacts (e.g. encrypted data, digital signatures, and message digests) with stronger algorithms and/or longer keys.

Document: A document can be any information which can be represented electronically or optically (e.g. an arbitrary bit stream).

Key Degradation/Algorithm Degradation: The process whereby the protection afforded a document by encryption under a key loses effectiveness over time. For example, due to factors such as improvements in computing speed and breakthroughs in cryptanalysis, it is expected that a document securely encrypted today would be crackable years later. This property could affect any cryptographic information, including digital signatures. This problem can be generalized to keyed and non-keyed cryptographic processes and artifacts, such as one-way hash algorithms. The security provided by these are also expected to diminish over time.

LTSV: Long Term Signature Verification. The herein described method and apparatus for verifying a digital signature after the certificates used for such verification have expired.

LTSV client: The entity which requests/utilizes the services of the LTSV server.

LTSV server: The entity which delivers the LTSV services. This does not imply, however, that this entity must be stand-alone component.

LTSV submission: A request from an LTSV client to

an LTSV server to perform the necessary functions required to enable reverification of a digital signature some time in the future (e.g. save PKI state).

PKI: Public Key Infrastructure. Refers to all components, protocols, algorithms, and interfaces required to deliver the capabilities to digitally sign and verify documents. For purposes of clarity herein, a PKI does not include a service module for long term signature verification (LTSV server), although in practice a PKI might be designed to encompass such a module.

Signature Reverification: The re-creation of the digital signature verification process after the original verification. This specifically refers to the process associated with the verification process, based upon the restoration of the previously saved PKI state.

Signature Verification: The process by which a digital signature, for a given document, is determined to be authentic or not.

Signature Verification Module: The module which is responsible for performing the verification of digital signatures.

Time stamp: A digital time stamp is an electronic indicator which associates the current date and time with a particular document. Time stamps are useful for proving that a document existed at a particular time. It is desirable that time stamps be secure, durable over time, and trusted by those using them.

The discussion herein assumes an understanding of public key, digital signatures, and PKI infrastructure using X.509 certificates. Practical information concerning application of such techniques is considered to be well known to those skilled in the art. Background information may be found, for example, in B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, Inc. (1996); W. Ford, M. Baum, Secure Electronic Commerce, Prentice Hall PTR (1997); and in the X.509 v.3 specification ([X.509-AM] ISO/IEC JTC1/SC 21, Draft Amendments DAM 4 to ISO/IEC 9594-2, DAM 2 to ISO/IEC 9594-6, DAM 1 to ISO/IEC 9594-7, and DAM 1 to ISO/IEC 9594-8 on Certificate Extensions, 1 December 1996). The system described herein may be built upon the X.509 infrastructure.

The following discussion provides some background on cryptographic techniques. Cryptographic algorithms can generally be divided into two categories: public key (e.g. RSA) and secret key (e.g. DES). Both types of algorithms transform plain text into cypher text using a key(s) for the encryption and decryption processes.

Both public key and secret key algorithms are considered to be secure. One is not better than another in terms of security. The strength of each algorithm, in terms of it being cracked, is largely a function of the length of the key used. The primary distinguishing characteristic of public key, however, is that it uses two keys (one to encrypt and another to decrypt), while secret key algorithms use only one key (the same key is used for

encryption and decryption). For this reason, secret key algorithms are sometime referred to as symmetric algorithms and public key algorithms are called asymmetric.

One problem with secret key algorithms is that a key must be distributed between all participants. This means that some secure channel must be available for the distribution of the keys.

In practice, each entity in a public key-based system has a key pair, i.e. one private key and one public key. The private key is known only to its owner, the public key is known to all correspondents. It is computationally infeasible to determine a private key from the public key.

The two primary services provided by public key cryptography are secure exchange of symmetric keys (by using public key techniques to encrypt a symmetric session key), and non-repudiation via digital signatures.

Public key cryptography can be used to solve the key exchange problem associated with secret key algorithms by using this technology to encrypt the secret key under the public key of the recipient. It can then be decrypted by the recipient using his/her private key.

Digital signatures are possible by encrypting data with the private key of the signing entity. Any entity can decrypt it with the signer's publicly available public key and know that no one else could have encrypted it because that private key is only known by that one individual. This particular use of public key provides the non-repudiation service, which is a primary use of public key cryptography. A digital signature is very powerful notion, it generally exhibits the following characteristics:

- Cannot be forged;
- Is independently verifiable;
- Is not reusable or transferable to a different piece of data; and
- Includes data integrity checks, allowing tamper-detection.

The new services provided by public key cryptography do not come for free, however, because these services require the existence of a supporting public key infrastructure. The strength of a public key system depends upon the assurance that all participants know the public key of any entity with whom they wish to correspond. If a secure correspondence between a user and his/her public key cannot be maintained, then it may be possible to impersonate another entity or read encrypted data intended for another.

The standard solution to this problem is the issuance of a digital certificate (X.509 certificate) to each participant. This certificate securely binds its owner's name with his/her public key. It is issued by a trusted third party, called a certification authority (CA), and is signed by that CA, thereby making it tamper proof. Certificates are issued for a limited period of time (start and

stop dates), during which the certificate is considered valid. A certificate is considered expired after the ending validity date.

The public keys of entities (which are embedded in the X.509 certificates) must be publicly available. The distribution or access mechanisms available are numerous.

The secure operation of a public key infrastructure rests upon certain points of trust. Certainly each entity must trust its own CA. However, when a given PKI domain is expanded to encompass relationships with multiple CAs, the number of points of trust are also expanded. The trust placed in a particular end entity (*i.e.* that entity's certificate or signature) is directly related to the trust relationships among the CAs which certify those entities.

CAs can create trust relationships with other CAs by certifying each other. This can be a unidirectional trust relationship, whereby one CA can merely issues a certificate to another CA, just as a CA issues a certificate to an end user. Two CAs can also mutually agree to trust each other (bidirectional trust relationship) by issuing a cross-certificate -- a special form of certificate which contain two individual certificates, one for each direction.

If two entities are in the same CA domain, then there is no concern with respect to CA trust because they both trust the same CA. This is not the case, however, when dealing with the scenario where entities which have been certified by different CAs attempt to conduct a secure transaction. The security of this transaction depends upon the trust between the CAs. More generally, the security provided by the PKI depends upon the trust models embodied in the trust relationships among the various CAs which choose to trust one another. In concrete terms, any change in these trust relationships can cause a signature verification to either succeed or fail.

The preferred method and apparatus effectively extend the time over which a digital signature can be verified, *i.e.* well beyond the expiration of any or all of the certificates upon which that signature depends. They can be used for any application domain where users want digital signatures to be used on long lasting documents (*e.g.* contracts), and be independently verifiable years or decades after signing the document. The preferred embodiment of the invention provides two alternative approaches to constructing a solution which delivers long term signature verification (LTSV).

Fig. 2 is a block schematic diagram illustrating a "save state" embodiment of the invention. This embodiment, largely entails the use of cryptographic information and techniques. Thus, an archive facility 20 is used to store a public key infrastructure (PKI) state 24, *e.g.* cryptographic information, such as certificates and CRLs, in addition to the source document itself. For example, the LTSV client 25 requests the services of an LTSV server 26 to accomplish storage of such information. This step is shown as the "save state" step in Fig.

2. The PKI state information may contain either or both of cryptographically protected information, such as certificates and CRLs, and information that is not cryptographically protected, such as the public key of a root certification authority or policy information.

This information comprises all that is necessary to re-create the signature verification process at a later time, *i.e.* during the "restore state" step, for example, as requested by the LTSV client. It may also be desirable to store the source document separately from the cryptographic information (such as the signature, certificates, and CRLs) for reasons of privacy. For example, a user may want to have control over the source document.

When a user wants to reverify the signature on a document, possibly years later, the LTSV server 26 re-creates the precise state of the PKI at the time the document was originally submitted. The LTSV server restores the state, and the signature verification process 28 executes the exact process it performed (or would have performed) years earlier. The time used as the basis for re-creation of the signature verification process does not have to be the time of submittal. Rather, the time could be some other relevant time, such as when a document was signed by the originator or when it was verified by a recipient.

Fig. 3 is a block schematic diagram illustrating a "save state" "secure storage" embodiment of the invention. This embodiment of the invention combines the strength of cryptography with the proven resilience of (non-public key) technology and procedures currently associated with secure data stores. An example of this embodiment:

- Saves the PKI state for future reverification (as described above in connection with Fig. 2); and
- Protects the PKI state information from intrusion by maintaining it in a secure storage facility which is protected by services, such as firewalls, access control mechanisms, audit facilities, intrusion detection facilities, physical isolation, and network isolation; and/or employing a cryptographic protection mechanism, for example using the LTSV server to sign the PKI state information or using a keyed hash algorithm.

In addition, other non-cryptographic features may be added to such approach to deliver a highly secure and trusted LTSV solution, including, for example utilities 30 for viewing the PKI state information (cryptographic as well as non-cryptographic) and visually monitoring the security of the system. These utilities can be used to provide visual evidence for purposes of dispute resolution.

One enhancement to the secure storage approach herein disclosed maintains certain evidence, such as certificate chains, in an archive. This information need

not be used for actual reverification, but merely as supporting evidence in case of a dispute. See A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, pp. 583 (1996), for one manner in which this enhancement may be implemented in the context of a notary service (discussed above).

There are other embodiments of the invention in which a hybrid LTSV solution could be constructed by combining cryptographic and non-cryptographic techniques. The best combination for a particular application domain depends upon the security requirements of the application(s), in combination with cost constraints.

It is presently preferred to employ the second embodiment of the invention (discussed above) due to the cryptographic strength associated with its ability to recreate the complete digital signature verification process, combined with the trust instilled by more conventional techniques used for providing secure storage, and in conjunction with audit and viewing facilities with which to view evidence and monitor the secure storage controls. In practice, the most useful embodiment of the invention for a particular application may be that which is the least expensive and which still meets the user or application requirements.

Several issues related to the design of a system which implements LTSV are described below. Alternatives for the resolution of the issues are presented, as well as a discussion of the advantages and disadvantages associated with each alternative. The best approach to any given solution depends upon the security requirements of the application(s) using the LTSV services, as well as the cost constraints. There is no best solution for all applications.

When to Save the PKI State

Signature reverification is preferably associated with a particular time because the outcome of this process could change, depending upon the state of the PKI (e.g. because of certificate revocations or the creation/removal of cross certificates). There are numerous possibilities with regard to when the PKI state should be saved, including:

- At signature creation time. This approach is used when an individual wants to document the validity of his/her signature at the time it was created. This is the most accurate time to store the PKI state because it reflects the state at the time of signing, which is presumably the critical time in evaluating the authenticity of that signature. Changes to the PKI state occur after that time, some of which could impact the outcome of a signature reverification. Therefore, saving of the PKI state at any time after signing introduces the possibility of inconsistencies between the signer's and recipient's perspectives on a signature's validity.

- At signature verification time. This approach is useful when a recipient wants to document the validity of a signed document received from another individual.
- At archival time. When a user decides that a document should be archived for long term storage is also an appropriate time to save the PKI state.
- When explicitly requested. There may occur certain application specific document life cycle milestones, at which time the user may desire the PKI state to be saved for future reverification.
- Just before changes in PKI state (e.g. certificate revocation). This approach requires a tight integration with the PKI because changes in the PKI must be monitored.

The correct time at which to save the PKI state is preferably determined by the constraints and needs of the application using the LTSV services. A robust LTSV solution is able to accommodate the needs of all (or most) applications in this respect.

Contents of the PKI State.

The exact composition of the PKI state varies somewhat from one PKI vendor's product to another's, depending upon the implementation chosen by each vendor. Moreover, certain information is stored in a different format from one vendor to another. In addition, the contents of a PKI state may change over time as well, as new capabilities (and supporting data) are added to the system. Finally, the required contents of the PKI state may change from one application to another, depending upon the needs (e.g. level of security and legal requirements) of each application.

Notwithstanding these uncertainties, there are classes of PKI state information which are candidates for saving. These classes include:

- Certificate chain (list of certificates from one entity to another, including certification authorities (CAs) and the end entities).
- CRLs (one for each CA in certificate chain).
 - CA policy statements or identifiers.
- Attribute certificates.
- Date and time.
- Trust information (e.g., public key(s) or certificate(s) of trusted root CA(s), policy constraints).

Policy constraints are, for example, non-crypto-

graphic information stored within the LTSV archive. The public key of the trusted root CA may or may not be cryptographically protected. If it is embedded in a certificate, then it is signed by the CA. However, it could just as well be an isolated public key, in which case it is unprotected by cryptography.

It is possible that the items in the above list may not be supported or available from certain PKI implementations. Further, the PKI state from another implementation might include some additional data. Therefore, the list above is only an example of what might be considered important pieces of PKI state information, given the current state of the technology. An implementation of an LTSV service is preferably tied to the implementation of a specific PKI until such time as the technology evolves and comprehensive standards emerge.

How to Store the PKI State

Storage of the PKI state is preferably accomplished in either of two general ways:

- Store all of the PKI state relevant to each document separately; and
- Store the PKI state centrally, and only store references to the PKI state information with each document. This approach enables storage efficiencies by eliminating the redundant storage of PKI state information over multiple documents. For example, given two documents submitted to the LTSV server at about the same time, it is possible that the CRLs contained in the PKI state are exactly the same for both submissions. Central storage of this information allows the LTSV server to store this information only once.

The storage requirements for the save state solution for LTSV may be quite large, depending upon the size of the certificates, the length of the certificate chains and -- more importantly -- the size of the CRLs. The choice of storage technique may have a great impact on the total data storage requirements. It is clearly undesirable to store massive CRLs with every document that is stored for long term archival and possible future reverification. For this reason, the second alternative listed above is presently considered to be the preferred approach.

However, this second approach may present certain difficulties in applications where the LTSV server is an entirely separate component from the PKI, and where support of multiple PKIs is a primary design goal of the LTSV server. In this case, it would be advantageous for the PKI state to remain opaque to the LTSV server, thereby providing ease of support of multiple PKI vendors. Given that what constitutes the PKI state for one vendor may be different for another vendor, it is desirable to maintain an opaque interface between the

LTSV server and the PKI. On the other hand, storage efficiencies can be derived only if the LTSV server is informed about the contents and format of the PKI state information. These conflicting requirements -- acceptable storage size and opaqueness -- pose a challenge for the design of an LTSV service.

Some of the possible alternatives are listed below:

- Keep the interface opaque and store the PKI state as it currently exists (full certificate chains and CRLs). This option focuses entirely on the opaqueness requirement, and sacrifices the data size requirement. The primary advantage of this solution is simplicity and quick deployment.
- Remove the opaqueness requirement by making the PKI state visible to the LTSV server. This allows the LTSV server to manage the certificates and CRLs manually -- thereby avoiding duplication of these objects in the data store. This solution potentially sacrifices the ease of multi-vendor support at the expense of achieving efficient storage.
- Compromise by making the CRLs visible to the LTSV server, where other PKI state information is opaque. This solution is interesting because it is probable that the CRLs are the largest piece of data comprising the PKI state. Because CRLs are standard across nearly all PKIs, the visibility should not pose a problem in terms of multi-vendor support. This solution address both of the requirements, but does put the burden of management of the CRLs onto the LTSV server.
- An alternative embodiment of the invention provides a variation on the solution above that breaks up the PKI state into multiple pieces, each of which is opaque. The PKI indicates which of these objects is common across multiple signed documents (e.g. CRLs and certificates). The PKI labels these objects with unique handles (identifiers), thereby allowing the LTSV server to store these objects and retrieve them efficiently when needed for signature reverification -- all the while maintaining the opaqueness of these objects.
- Encourage PKI vendors to make concise cryptographically protected assertions about the state of revocation, as an alternative to using CRLs. (For example, CRLs indicate who has been revoked. It would be more efficient if the PKI could make a statement that a certificate has not been revoked at a given point in time. This could eliminate the need for storing CRLs.) This approach is non-standard, but acceptable because these PKI-generated assertions are not seen by any application outside the PKI. A major benefit of this approach is that the opaqueness of the state is preserved while some of

the storage inefficiencies of the state information are removed.

For cases where the LTSV server is dedicated to a particular PKI, it is preferred to create a close integration between the two components, thereby allowing the LTSV server to know about the content and format of the PKI state information, and store it in the most efficient manner possible. For cases where the LTSV server must be insulated from the PKI (*e.g.* for portability across multiple PKIs), one of the options listed above (with the possible exception of the first two) may be used.

Location of Source Data.

The source data associated with an LTSV submission can be stored either by the client or by the LTSV server itself. Some LTSV clients do not choose to submit clear text to the LTSV server for storage because of concerns over privacy. (Privacy of the channel between the LTSV client and the LTSV server can be achieved by having the client encrypt the submission under the public key of the LTSV server.) A submission to the LTSV may be encrypted, such that the LTSV is not able to decrypt it. That is acceptable with the LTSV server. However, the client must determine how to decrypt the submission.

Given that the LTSV server views the source data as a bit stream, it is possible that the message could be encrypted by the LTSV client before submission. (The fact that a general purpose LTSV server treats the source document as a bit stream does not preclude the possibility of implementing an application specific LTSV server that is aware of the contents of the submitted data.) The LTSV server treats the encrypted data as the source. Such prior encoding may be sufficient for some applications' needs for privacy. In this case, however, either the client must maintain the decryption key, or the key must be divulged and stored by the LTSV server (which is probably not acceptable).

Alternatively, the LTSV client may submit a message digest (resulting from a one-way hash function) as the source document. The client, in this case, is responsible for maintaining the real source document. If the source document is stored by the client, then only the PKI state information is stored in the LTSV server's archive (along with some reference to the source document or the submitter).

Whether the source data is stored by the client or the LTSV server, it must be produced if and when a reverification of that document is required. It is a required component of any signature verification process.

Key and Algorithm Degradation.

If cryptographically encoded information (*e.g.* digital signatures or encrypted data) is stored for a significant

period of time, the issue of key and algorithm degradation must be addressed, *i.e.* the probable loss in effectiveness of a cryptographic key or algorithm over time. Although signed documents are expected to be sealed securely with strong cryptographic algorithms, the strength of an algorithm and associated key length decreases over time with the advent of faster computers and new developments in cryptanalysis. It is expected that cryptographic algorithms and key lengths have limited life spans. It is generally acknowledged that they should be examined, modified, and/or replaced at periodic intervals. This legitimate security concern increases with the length of time for which a document is valid, and it becomes a very serious threat as the time span approaches multiple decades.

For example, a digital signature performed today, using RSA and a 512-bit key, is considered very strong (*i.e.* it would take years to forge it). But, it is also expected that this same signature may be easily forgeable within ten years or so. This is because of the increased ability to search the key space faster (and thereby find the key used to sign the message) with newer computers or computing techniques. Similarly, there may continue to be developments in techniques for factoring large prime numbers (the difficulty of which is the basis for the strength of the RSA algorithm). It is reasonable for both of these abilities to improve over time (although the pace of these changes is less certain).

It is, therefore, prudent to protect cryptographically encoded documents from this threat when those documents must live beyond a few years. This is the case with the documents expected to be submitted to the LTSV server, and especially so when using the save state approach herein disclosed. Hence, the LTSV facility should address this problem. Not only must the signed documents stored in the archive be protected from this threat, but all other cryptographic data or metadata stored in the archive should be protected. (The cryptographic data primarily include keyed information. That is, any information that is signed or encrypted with a private key. Such information may also include non-keyed cryptographic data, such as the output from a hash algorithm, such as MD5.) This data could also include such items as certificates and CRLs, which are, themselves, digitally signed by the issuing CA.

There are a number of ways that the LTSV facility addresses this problem. For example:

- Periodically countersign all data in need of cryptographic refresh through the use of nested signatures. Under this approach, the LTSV server effectively refreshes the cryptographic strength of the data by signing it with successively longer keys (or stronger algorithms) every few years. Each counter signature has the effect of locking in the cryptographic strength of the enclosed signature(s), thereby extending the cryptographic life of the enclosed document. This countersignature is prefera-

bly performed by the LTSV server using a key owned by that server. Performance shortcuts may be required to avoid the costly unraveling of signatures at reverification time, or the potentially time consuming task of countersigning every document in the archive. Such shortcuts include, for example, removing a previous countersignature before applying a new one, or countersigning the entire archive or portions thereof instead of each individual document.

- A modification of the cryptographic approach suggested above provides for countersigning the information in the archive once, but with an extremely long key, *i.e.* a key which is expected to be unbreakable for decades or more. This eliminates all need for countersigning. This may be merely a theoretical solution because finding an algorithm and key length which is secure for that long is impossible to predict. Therefore, there is still a need to provide some backup mechanism, just in case the original algorithm were cracked, for example.
- Protect the cryptographic information in the archive by insulating the archive itself, rather than the individual documents contained in the archive, thereby eliminating the need for a cryptographic solution. In this approach, the archive is protected via access controls and other procedural controls. If the archive can be effectively insulated from intrusion and modification, then key degradation is not an issue and cryptographic refresh is not necessary.
- Use a time stamp facility to seal the cryptographic information in time. Such a facility is expected to provide all of the necessary characteristics required for solving the key degradation problem. This time stamp facility could use one of the techniques listed above, or it could even be an independent service (see below for a discussion of time stamping).

Relationship to Time stamping.

A secure and comprehensive LTSV solution preferably includes an association with a time stamping mechanism. For long term verification of digital signatures, it is often necessary to know the time at which particular events occurred (*e.g.* time of signing or verifying a signature) to determine if a document was valid at that specific time. If there were uncertainty concerning when a document was signed, then the later reverification of that document could be compromised because of the uncertainty of when it was signed.

Fig. 4 is a flow diagram that provides two alternative scenarios that illustrate the applicability of time stamps.

In scenario 1:

- Alice signs a document at time T1, and sends it to

Bob (140).

- Alice's certificate is revoked at time T2 (142).
- Bob verifies Alice's signature at time T3 (144).

In scenario 2:

- Alice's certificate is revoked at time T1 (150).
- Alice signs a document at time T2, and sends it to Bob (152).
- Bob verifies Alice's signature at time T3 (154).

When Bob performs the verification (at time T3), he does not know when Alice signed the document. This is critical, because if Alice's key (certificate) were revoked before signing the message, then the signature verification by Bob should fail, and Bob should not trust the contents of the message. If, on the other hand, the revocation occurred after the act of signing, then the signature can be presumed to be valid and trustworthy. For simplicity, this example does not consider the complicating issue of CRL latency, *i.e.* the time between the initiation of certificate revocation and the time when this information becomes available on a CRL.

This example demonstrates the need for a secure and trusted time stamp mechanism in the domain of digital signatures. The mere recording of the current date and time when creating a digital signature is not sufficient for most application because the source of that time may not be trusted by the recipient. The impact, however, also applies not only to the short term signature verification process, but also to the long term verification of digital signatures. Given the example above, the LTSV server could save the PKI state (at time T1) associated with scenario 1 or scenario 2 (or both). The outcome of a signature verification on this message years later is greatly affected by the PKI state used for this verification process, as well as the target time for the verification.

The problem highlighted above demonstrates the preference that the LTSV service to be cognizant of time. It should:

- Be able to determine in a secure fashion the time at which a document was originally signed;
- Be able to re-create accurately the PKI state which was active at a target time in the past;
- Be able to determine the current date and time accurately; and
- At a minimum, save the PKI state associated with a particular target time.

These requirements establish the preference for the integration of a time stamp facility with the signing and verification (and reverification) process. When a document is signed, it is also preferably time stamped to document in a secure fashion the precise moment at which that event occurred. The LTSV service should know the time for which the PKI state is to be saved, be sure to save the appropriate state (the state active at the target time), and execute its signature reverification process in the context of the correct time.

Usage Scenarios.

Figs. 5a-5c provide block schematic diagrams that illustrate three long term signature verification usage scenarios.

In scenario 1, a client (EntityA) 50 submits a document to a LTSV facility 52 for long term signature verification. This is a simple case where EntityA is interested in documenting that it possessed some piece of information.

In scenario 2, EntityB 56 receives a document from EntityA 54 and submits that document to the LTSV facility 58. In this case, EntityB wants to document that it received some information from EntityB.

In scenario 3, EntityA 60 sends the same document to EntityB 64 and to the LTSV facility 62. This case represents a carbon copy feature, whereby EntityA can document the information it sent to EntityB by additionally filing it with the LTSV facility.

Each of the scenarios described above raises issues with respect to encryption, private key access, and trust models.

Encryption and Private Key Access.

A document can be encrypted and/or signed. Ideally, the LTSV facility accepts any such document. This raises a problem, however, with respect to how the LTSV module works with respect to the encryption. When encrypting under a public key system, the document is effectively encrypted under the public key of the recipient, thereby guaranteeing that the recipient (the possessor of the corresponding private key) is the only entity which can decrypt the information. (For purposes of this discussion, interaction with symmetric keys and algorithms is ignored.)

When applying this principle to scenario 1, it is clear that if the signed message is also encrypted, then it could be encrypted under the public key of the LTSV module. This allows the LTSV component to unwrap the signed document and preserve it for long term verification. This is a useful feature because it provides confidentiality between EntityA and the LTSV service. This scenario does not preclude the possibility that the source document sent signed and encrypted to the LTSV module could itself be encrypted under a key known only to EntityA. That is, it is not necessary that

the LTSV have access to the plain text version of the source document. The LTSV module treats that encrypted document as the source. If EntityA does decide to encrypt the document first under a secret key before submitting the document to the LTSV service, then it is the responsibility of EntityA to maintain possession of that key if and when decryption of that document is required.

In Scenario 2, if the message from EntityA to EntityB is encrypted (under the public key of EntityB) and then forwarded -- unchanged -- to the LTSV service by EntityB, then it is unreadable by the LTSV component because it does not possess the private key required to decipher and unwrap the enclosed signed document. This unwrapping (decipherment) is essential for the LTSV module to do its job.

There exist several alternatives for addressing this problem:

- Allow the LTSV facility to have access to EntityB's private key;
- Do not allow EntityA to send encrypted documents to EntityB; or
- Have EntityB strip off the privacy aspect of the signed and encrypted document received from EntityA. Because EntityB wants to preserve EntityA's signature on the document, and be able to verify it at a later time, this stripping process can not alter the validity of the signature. EntityA can then either send the stripped (*i.e.* plain text) document to the LTSV service, or it can re-encrypt it (still preserving the original signature by EntityA) under the public key of the LTSV module.

The latter approach above is presently the preferred approach. The first approach above raises significant security concerns because it requires distribution of an entity's private key. The second approach above is unacceptably restrictive on the usage of the system.

Trust.

Digital signature verification is always performed between two (and only two) entities. The verification process is based upon (among other things) the trust relationship(s) in place between those two entities -- the originator (signer) and the recipient (verifier).

Fig. 6 is a block schematic diagram that illustrates trust between two entities according to the invention. In this situation, EntityA 70 has been issued a certificate by CA1 72, EntityB 74 has been issued a certificate by CA2 76, and CA's 1 and 2 have been cross certified. (A cross-certificate is a special type of certificate which indicates mutual trust between two CAs.) The resulting trust model sets up a path of trust between EntityA and EntityB, enabling them to verify digitally signed docu-

ments from one another successfully. (A trust model is comprised of the trust relationships among PKI entities (CAs and end users), embodied by the certificates and cross-certificates issues among these entities, as well as the underlying policies which enable this trust.) Note that if any of the three paths in this model were not in place, then sufficient trust would be lacking for the successful exchange of digitally signed messages between the two end parties. Signature verification would fail if any entity in this path is not trusted.

This trust path is commonly referred to as the certificate chain because it is composed of the certificates between the two entities. When considering the save state approach to long term signature verification, it is this entire trust path (among other things) which must be archived as part of the PKI state for later signature reverification. Moreover, the trust path stored by the LTSV facility must contain the relevant trust information existing at the time of the request, not at some other time (before or after) where the trust relationships may be different between the entities. For example, a cross certificate between to CAs could either be created or removed at some point in time. This could effect the trust between two entities and affect the outcome of a signature verification.

As discussed above, the time associated with the existing trust model between two entities is extremely important to the LTSV facility, but there are also ramifications with respect to how the LTSV module works -- specifically, what trust information is needed and stored by the LTSV component for later signature verification. This gets complicated when the LTSV component is included, which may or may not be trusted (via some trust path) by some entities.

Consider the three scenarios illustrated in Figs. 5a-5c:

Scenario 1 is fairly straightforward. There are only two entities involved. The trust path stored by the LTSV facility is the path between those two parties (EntityA and LTSV). It is assumed that trust exists between these entities, otherwise EntityA would not submit a request to that service.

Scenario 2, however, raises certain issues. When EntityB sends a request to the LTSV service, what signature does EntityB want to later verify? Most likely, EntityB wants to reverify EntityA's signature at a later time -- it wants the LTSV service to document that the signed document received from EntityA was valid (contained a valid signature) at the time it was received. This raises two general questions:

- Whether the LTSV service is trusted by EntityA. It can be assumed that the communicating parties (EntityA with EntityB, and EntityB with the LTSV) have developed some trust between themselves. But in this case, it is possible that there exists no trust path between EntityA and the LTSV component.

- The trust path that is to be stored by the LTSV facility. There exist three possible trust paths which can be stored by the LTSV, *i.e.* the path between Entities A and B; the path between EntityB and the LTSV component itself; and the path between EntityA and the LTSV component, if it exists.

Fig. 7 is a block schematic diagram that illustrates a long term signature verification trust model. Given scenario 2, where EntityB 84 submits a signed document, received from EntityA 80, to the LTSV component 88, the LTSV can save the trust model embodied in the original signed document (EntityA 80 → CA1 82 → CA2 86 → EntityB 84). Later verification of this signature recreates the verification process originally performed by EntityB when it received this document from EntityA. If, however, the PKI state stored by the LTSV service were to contain only the trust path between the submitter and the service (EntityB 84 → CA2 86 → CA3 90 → LTSV 88), then reverification of the original document, as originally performed, is impossible. In fact, this is exactly the paradigm described in scenario 1, where the trust path between the submitter and the LTSV are of interest.

The above discussion reveals that there are good reasons for the LTSV component to be able to store either trust path, depending upon the requirements of the client.

In scenario 2, the LTSV would most likely store the trust path corresponding to the message from EntityA to EntityB (to reverify the signed document from EntityA to EntityB). In scenario 1, the LTSV would store the trust path corresponding to the submission itself -- from EntityA to the LTSV.

Similarly, scenario 3 represents a case where flexibility in which trust path(s) to store is required. In this case, EntityA's submission to the LTSV facility may be with the intent to either reverify its correspondence with EntityB, or to reverify the submission itself (between EntityA and the LTSV). In fact, both trust paths may be of use to the client. The requirements on the LTSV are determined by the business of the particular application being deployed. For this reason, the interface to the LTSV preferably supports the ability of the client to indicate the needs in terms of trust paths as it impacts the requirements for later reverification.

The disclosures in United States patent application no 08/892,792, from which this application claims priority, and in the abstract accompanying this application are incorporated herein by reference.

Claims

1. A method of enabling long term verification of digital signatures, comprising the steps of:

submitting a source document or digest thereof to a signature verification entity; and

using an archive facility to store a public key infrastructure (PKI) state relative to said document at a selected archival time.

2. A method as in claim 1, comprising the steps of: 5

using said archived PKI state to re-create said PKI state relative to said document at a selected time after a certificate associated with said signature has expired; 10
wherein the time over which a digital signature associated with said document can be verified is extended beyond expiration of any or all of any certificates upon which that signature depends. 15

3. A method as in claim 1 or 2 comprising the step of: storing said source document separately from any associated cryptographic information. 20

4. A method as in claim 1, 2 or 3 wherein the selected archival time used as the basis for subsequent re-creation of a signature verification process is the time of said source document submittal; 25
is the time when said source document was signed by its originator; or in the time when said source document was verified by a recipient.

5. A method as in any preceding claim, comprising the step of; 30
protecting said PKI state information from intrusion by maintaining it in a secure storage facility preferably comprising of at least one of a firewall, access control mechanism, audit facility, intrusion detection facility, physical isolation and network isolation; or 35
protecting non-cryptographic PKI state information from intrusion by protecting it in an archive via any of a signature and keyed hash algorithm.

6. A method as in any preceding claim comprising the step of: 40
providing utilities for viewing said PKI state information and for visually monitoring system security.

7. A method as in any preceding claim, wherein classes of PKI state information may include one or more of certificate chain from one entity to another, including certification authorities (CAs) and the end entities; certificate revocation lists (CRLs), one for each CA in certificate chain; certificate practice statements; attribute certificates; policy constraints; trust information; and date and time. 50

8. A method as in any preceding claim, comprising the step of: 55
periodically countersigning all data in need of cryptographic refresh through the use of nested signatures and/or countersigning information in said ar-

chive facility once with an extremely long key.

9. A method as in any preceding claim, comprising at least one of the steps of:

protecting said archive facility itself, rather than individual documents contained in said archive; and
employing a cryptographic protection mechanism at said signature verification entity.

10. A method as in any preceding claim, comprising the step of:
using a time stamp facility to seal cryptographic information in time.

11. Apparatus for long term verification of digital signature, comprising:

a source document; and
an archive facility for storing a public key infrastructure (PKI) state relative to said document at a selected archival time.

12. Apparatus as in claim 11, comprising:
either of a signature and a keyed hash system for protecting non-cryptographic PKI state information from undetected modification, wherein said noncryptographic PKI state information is maintained in an archive.

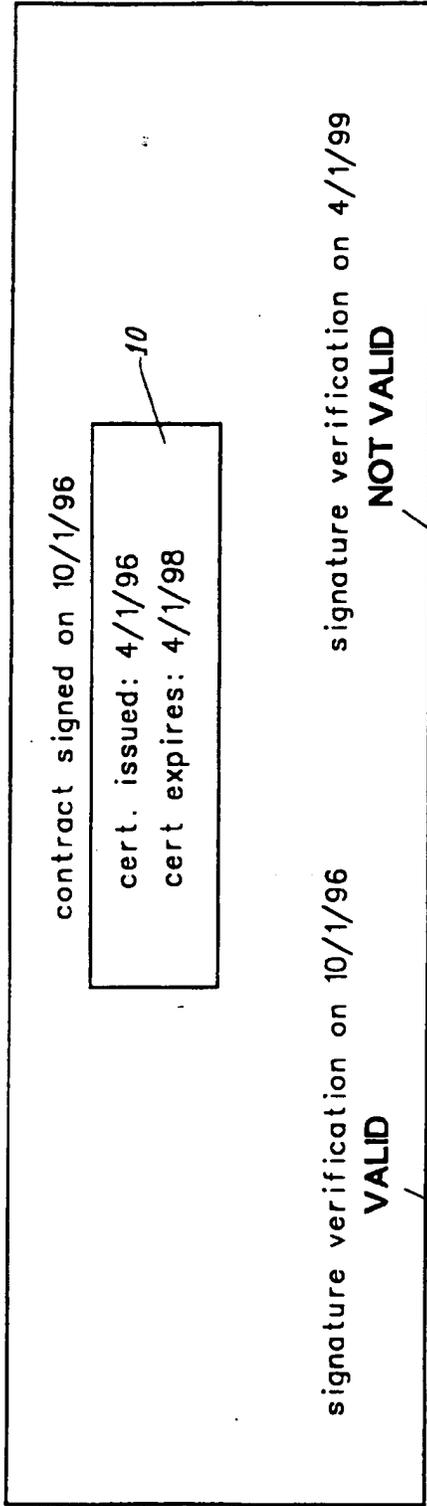


FIG. 1 (PRIOR ART)

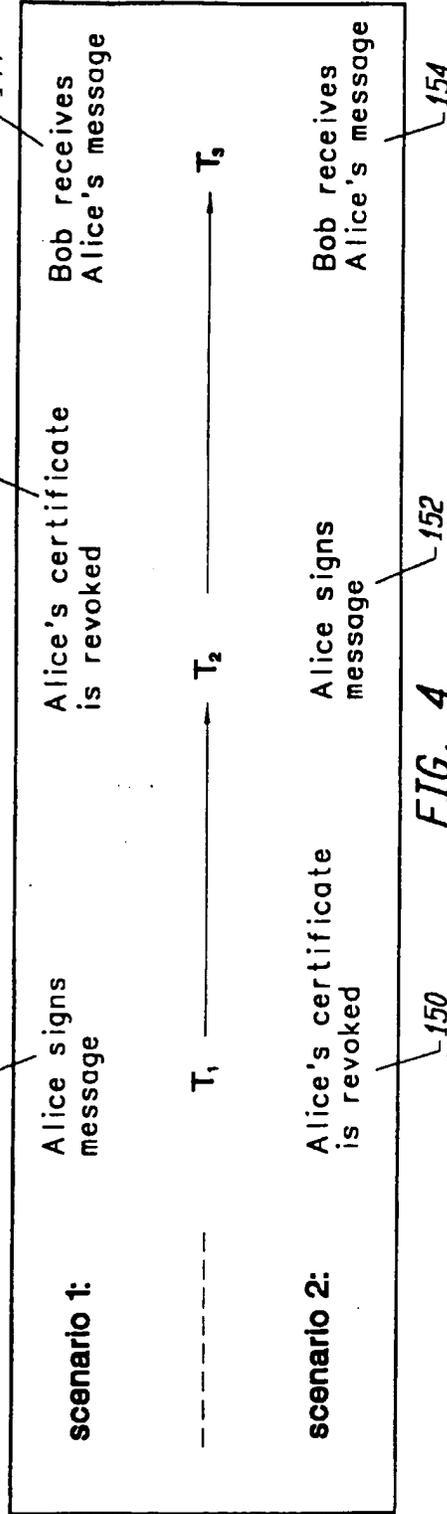


FIG. 4

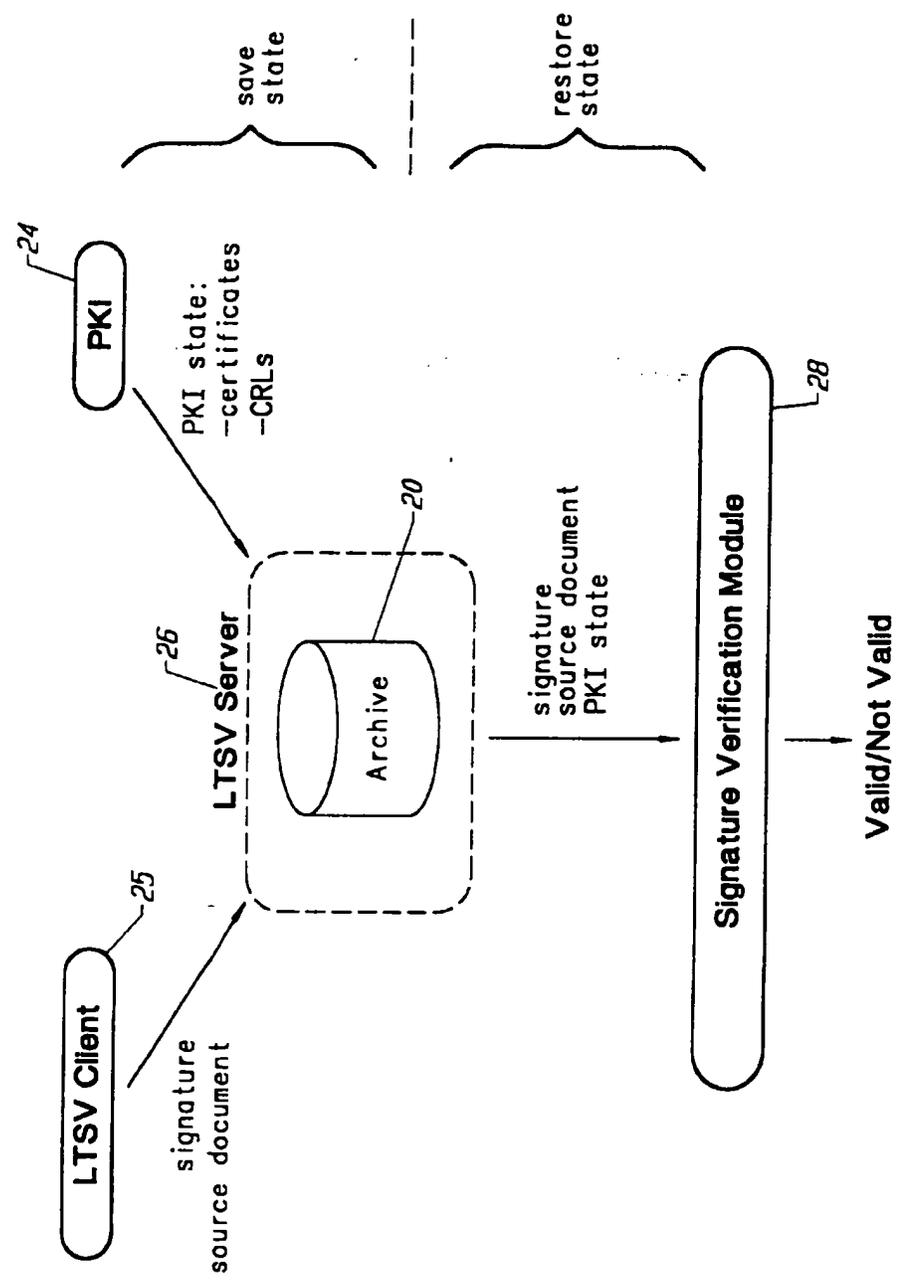


FIG. 2

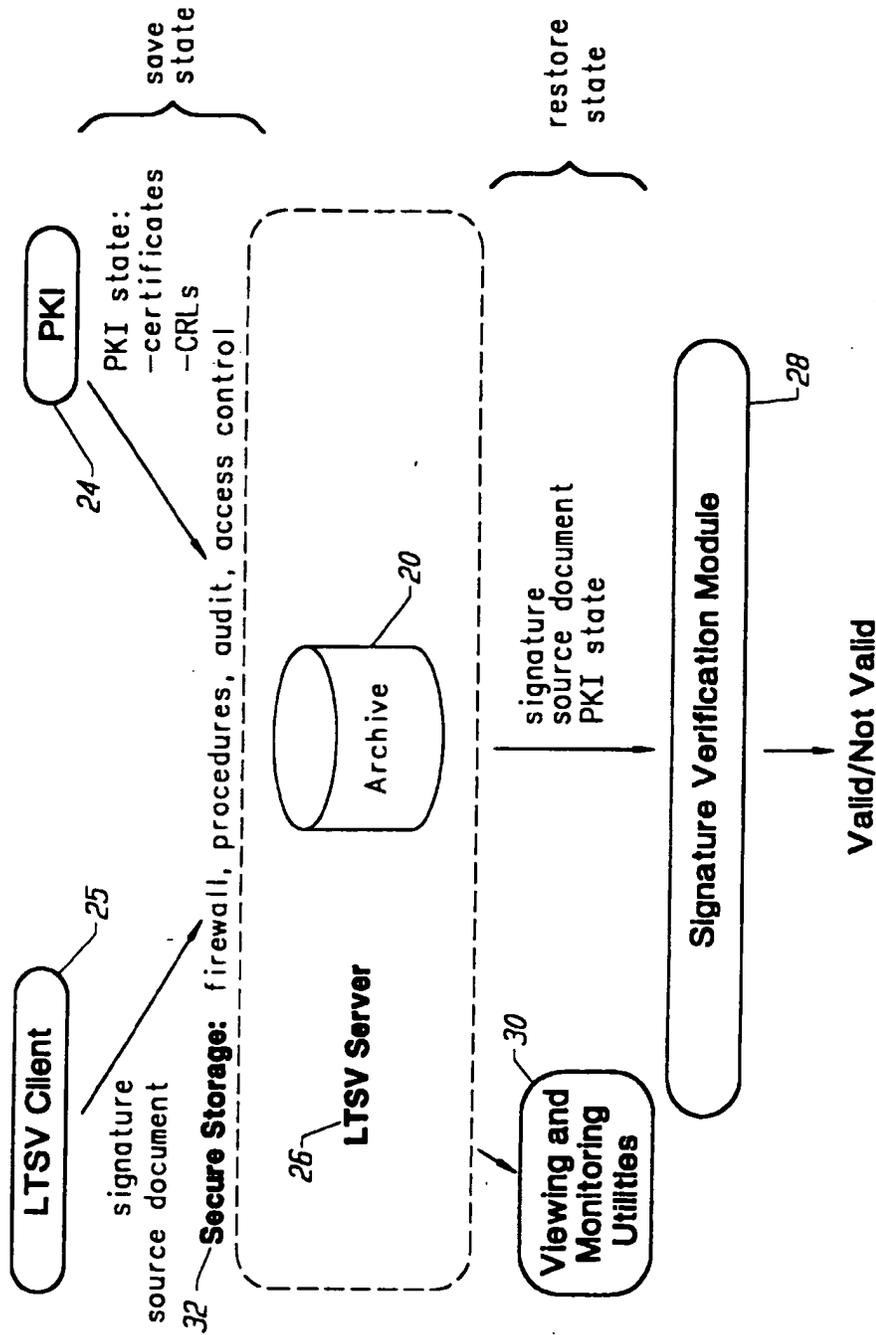
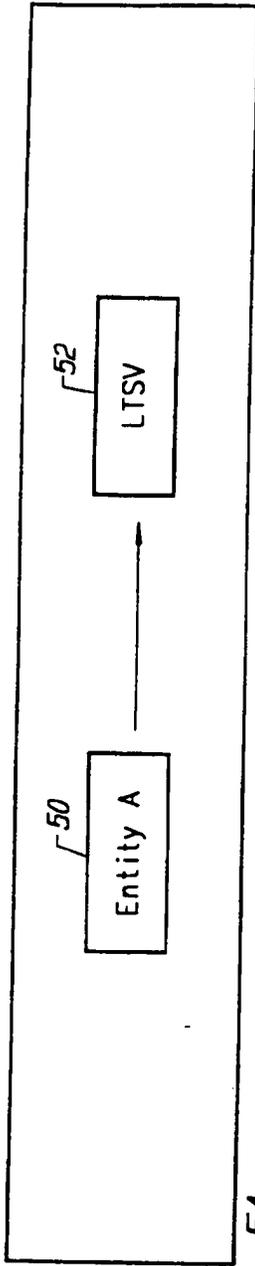
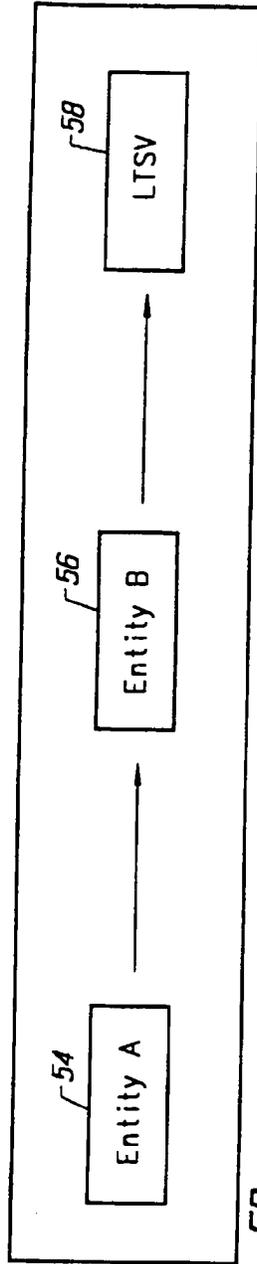


FIG. 3



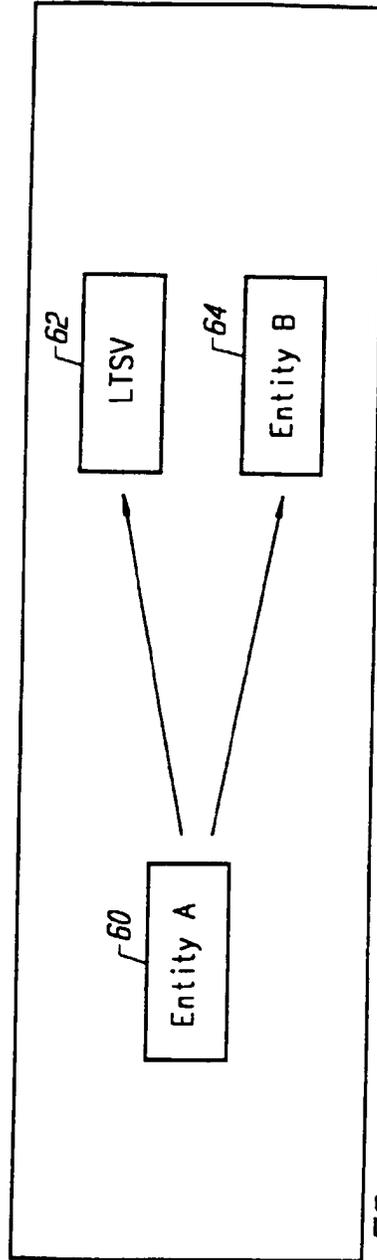
SCENARIO 1

FIG. 5A



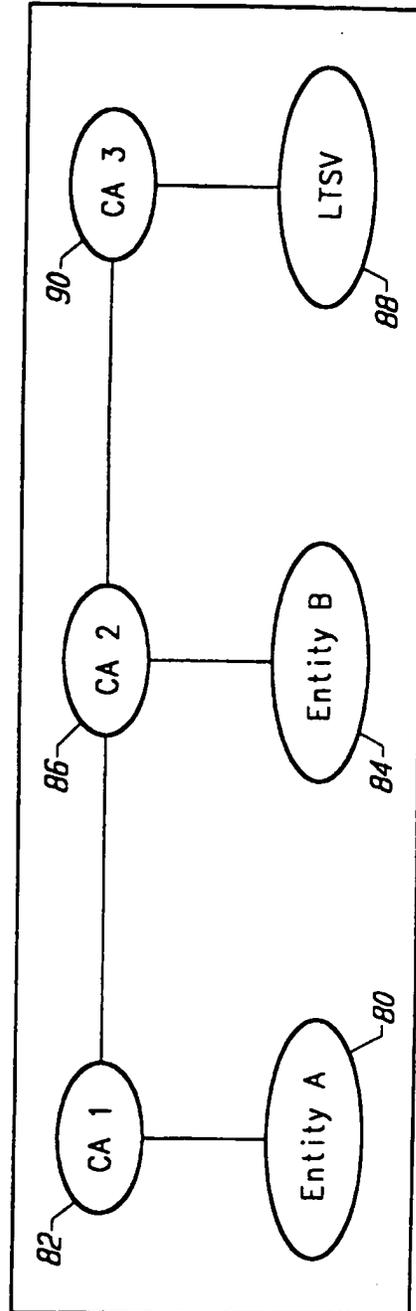
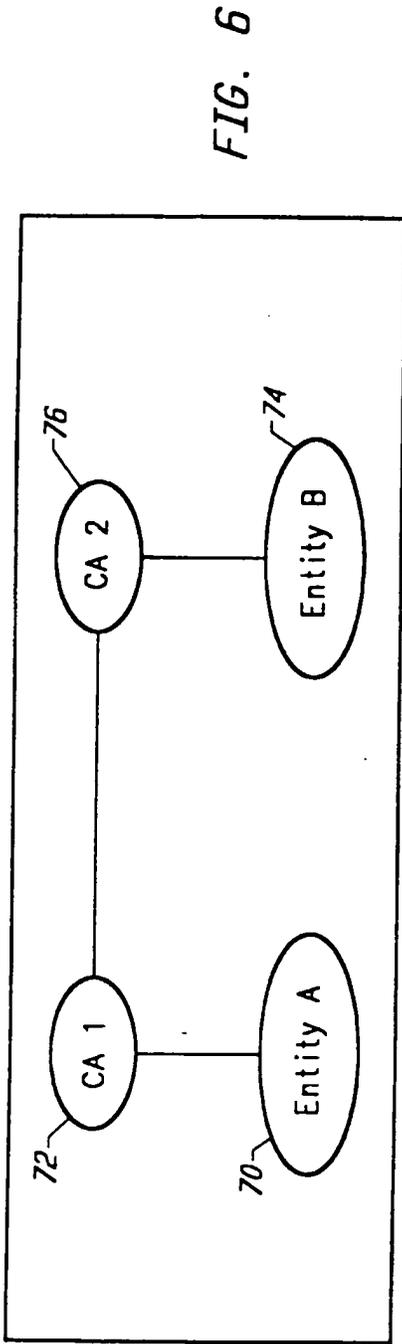
SCENARIO 2

FIG. 5B



SCENARIO 3

FIG. 5C



PATENT SPECIFICATION

(11) 1 483 282

1 483 282

- (21) Application No. 52131/74 (22) Filed 2 Dec. 1974
- (31) Convention Application No. 7342706
- (32) Filed 30 Nov. 1973 in
- (33) France (FR)
- (44) Complete Specification published 17 Aug. 1977
- (51) INT CL² G06F 13/00
- (52) Index at acceptance

G4A 10EX 13E 13M 17B4 17P 6G 6H 6X AP ND NR



(54) APPARATUS FOR PROTECTING THE INFORMATION
 IN AN VIRTUAL MEMORY SYSTEM
 IN PROGRAMMED DATA PROCESSING APPARATUS

(71) We, COMPAGNIE INTERNATIONALE POUR L'INFORMATIQUE CII-HONEYWELL-BULL, (formerly Compagnie Honeywell-Bull), a French Body Corporate, of 94 Avenue Gambetta, Paris 75020, France, do hereby declare the invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:—

The present invention concerns apparatus for protecting the information in a virtual memory system in programmed data processing apparatus.

Several schemes have been utilized in the past in order to protect information. Some of them are detailed by Robert M. Graham in a paper entitled "Protection in an Information Processing Utility", published in CACM (May 1968).

This type of memory protection is inadequate for present day multiprogramming systems because there is no provision for gradations of privilege or gradations of accessibility, and severely limits the control over access to information. There should be provisions for different access rights to the different types of information. A partial answer to this problem is found in the concept of a memory having a segment as the unit of information to which access is controlled (see Patent Application No. 21630/74, (Serial No. 1,465,344), filed on 15 May 1974). Varying degrees of access to each segment is possible by providing for different types of privileges attached to each segment such as master/slave, write/no-write and execute/non-execute. However, this method of protecting the privacy and integrity of information does not take into account the user of the information. Under this type of protection, privilege is not accorded the user but the information being protected. Hence a user if he has access at all to a segment has access similar to all other users who have access to the segment. David C. Evans and Jean Yves LeClerc in a paper entitled "Address Mapping and the Control of Access in an Interactive Computer," SJCC 1967, recognized the problem and attempted a solution. Evans and LeClerc said in that article p. 23, "The user of a computing system should be able to interact arbitrarily with the system, his own computing processes, and other users in a controlled manner. He should have access to a large information storage and retrieval system called the file system. The file system should allow access by all users to information in a way which permits selectively controlled privacy and security of information. A user should be able to partition his computation into semi-independent tasks having controlled communication and interaction among tasks. Such capability should reduce the human effort required to construct, debug, and modify programs and should make possible increased reliability of programs. The system should not arbitrarily limit the use of input/output equipment or limit input/output programming by the user". Evans and LeClerc proposed conditioning access rights on the procedure-in-execution. The segment, under their proposal, is still the unit of information to which access is controlled; however, a segment's access control attributes are recorded substantially in a user-name versus procedure tables whose entries are the access modes. Such a solution, however, has serious drawbacks. For one, the construction and updating of each segment's table of access control attributes presents a formidable task. For another, too many uses of the segment and event occurrences must be foreseen. To overcome this problem access control by procedure-set was suggested. Under this suggestion, related procedures are grouped into "sets of procedures" and access rights to segments is based on the identity of the set to which the procedure seeking access

belongs. This method alleviated the problem of constructing and updating each segment's voluminous tables of access control attributes, but introduced the problem of determining to which set a given procedure belonged, particularly when a procedure was or could be a number of many sets. This ambiguity in defining sets, and the possible transitions between sets makes the implementation of access control based on "sets of procedures" extremely difficult.

To overcome the difficulties encountered with the "set" technique a ring concept was developed. The ring concept groups the sets of procedures into rings that can unambiguously be ordered by increasing power or level of privilege. By assigning a collection of sets to a collection of concentric rings, and assigning numbers to each ring with the smallest ring having the smallest number and each succeeding larger ring having a progressively greater number, different levels of privilege can then be unambiguously assigned to the user of a segment. Under this concept the innermost ring having the smallest number assigned to it has the greatest privilege. Hence it can be postulated that users in the lowest ring number can access information having higher ring numbers, but users in a higher ring number cannot access information having lower ring numbers or can access information in a lower ring number only in a specified manner. This palpable change of power or level of privilege with a change in rings is a concept which overcomes the objections associated to a change of sets.

Multics (*Multiplexed Information and Computing Service*) is an operating system developed primarily by Massachusetts Institute of Technology, in cooperation with General Electric Co. and others which first utilized the ring theory of protection in software on a converted Honeywell 635 (Registered Trade Mark) computer and later on a Honeywell 645 (Registered Trade Mark) computer. The Multics philosophy utilizes 64 rings of protection numbered as rings 0-63 and is set forth generally in a paper entitled "Access Control to the Multics Virtual Memory" published by Honeywell Information Systems Inc. in the Multics Technical Papers, Order No. AG95, Rev. O. A more detailed description of Multics ring protection is to be found on chapter 4 of a book entitled "The Multics System: An Examination of its Structure", by Elliott I. Organick, published by MIT Press, and also in the Multics System Programmers Manual 1969, MIT Project MAC. Briefly, the Multics system does not utilize a "pure ring protection strategy" but rather employs the "ring bracket protection

strategy" wherein a user's access rights with respect to a given segment are encoded in an access-mode and a triple of ring number (r1, r2, r3) called the user's "ring brackets" for a given segment. A quotation from pages 137-139 from the Multics Technical Paper entitled, "Access Control to the Multics Virtual Memory" sets out the rules and conditions for using and changing rings.

This "ring protection concept" was first implemented with software techniques utilizing 64 separate rings. Subsequently an attempt was made to define a suitable hardware base for ring protection. The Honeywell 645 (Registered Trade Mark) computer represents a first such attempt. The Honeywell 645 (Registered Trade Mark) system differs from the "ringed hardware" concepts described supra in several respects which when taken together, add up to the fact that the Honeywell 645 (Registered Trade Mark) is a 2-ring rather than a 64-ring machine, and has in lieu of a "ring register", a master mode and a slave mode, which imparts greater power to the processor when in master mode than when in slave mode. "The access control field of the 645's SDW (segment descriptor word) contains no information about rings; in particular it does not contain ring brackets. It does, however, contain either:

- a) access-mode information possibly including either of the two descriptors; accessible in master mode only, master mode procedure;
- b) the specification of one of eight special 'directed' faults (traps) which is to occur whenever the segment descriptor word (SDW) is accessed.

"The procedure is only 'in master mode' when executing a procedure whose SDW indicates a 'master mode procedure'. The processor may enter master mode while executing a slave mode procedure by: faulting, taking an interrupt".

"The 645 processor's access control machinery interprets the SDW during the addressing cycle and causes the appropriate action to occur depending on the SDW and (usually) on the attempted access, as follows:

- a. If the SDW implies a particular "directed fault", then that fault occurs.
- b. Otherwise, if the SDW does not permit the attempted access, the appropriate access violation fault occurs.
- c. Otherwise, the SDW permits the attempted access and the access is performed.

"When a fault occurs, the 645 enters master mode and transfers control to the

70

75

80

85

90

95

100

105

110

115

120

125

appropriate master mode fault handling procedure". (Access Control to the Multics Virtual Memory, supra pps. 157—158).

5 Another paper by Michael D. Schroeder and Jerome H. Saltzer entitled "A Hardware Architecture for Implementing Protection Rings" published in Communications of the ACM, March 1972 Vol. 15, No. 3, sets forth background and theory of ring protection and describes a hardware implementation of "ring protection".

10 Because the Multics and Honeywell 645 version of ring protection was implemented mainly in software, considerable operating system supervisor overhead was entailed particularly when calls to greater or lesser power were made by trapping to a supervisor procedure. What was required was an access control mechanism which had the functional capability to perform effectively its information protection function, was relatively simple in operation, was economic to build, operate and maintain, and did not restrict programming generality. The Honeywell 6000 (Registered Trade Mark) computer system met these requirements by implementing most of the ring protection mechanism in hardware. Hence special access checking logic, integrated with the segmented addressing hardware was provided to validate each virtual memory reference, and also some special instructions for changing the ring of execution. However certain portions of the ring system particularly outward calls and returns or calls to a lesser power and returns therefrom presented problems which required the ring protection function to be performed by transferring control to a supervisor. What is now needed are further improvements in hardware and techniques that will permit a full implementation of ring protection in hardware/firmware and will meet the criteria of functional capability, economy, simplicity and programming generality.

40 Accordingly the present invention has for an object to provide an improved computer ring protection mechanism.

45 Accordingly the present invention consists in an internally programmed data processing apparatus CPU having a virtual memory system, and being responsive to internally stored instruction words for processing information and having stored in said virtual memory system a plurality of different types of groups of information each information group-type associated with an address space bounded by a segment having adjustable bounds, and comprising means for protecting the information in said-virtual memory system from unauthorized users by restricting

accessability to the information in accordance to levels of privilege, said means comprising in combination with an access checking mechanism:

(a) first means arranged in operation to store in said virtual memory system at least one segment table comprising a plurality of segment descriptors with each segment descriptor being associated with a predetermined one of said segments and each segment descriptor having a predetermined format containing an access information element and a base address element in predetermined positions of said format, said base address element being used for locating in said virtual memory system the starting location of a selected one of said segments, and said access information element for specifying the minimum level of privilege required for a predetermined type of access that is permitted in a selected one of said segments;

(b) a plurality of second means having a predetermined format, communicating with said first means, arranged to store in a predetermined portion of said second means, a segment number SEG for identifying a segment table and the location of a segment descriptor within said segment table, said second means also being arranged to store in a predetermined other portion of said second means, an offset address within the segment identified by said segment descriptor said offset address locating from said segment base the first byte of a word within said segment;

(c) third means responsive to an address syllable element of an instruction being executed for addressing one of said plurality of second means;

(d) fourth means arranged to store a displacement from said address syllable;

(e) fifth means, communicating with said first, second, third and fourth means, arranged to add the displacement D and said base address to said offset; and,

(f) sixth means responsive to said access information element in a selected one of said segment descriptors, restricting the accessability to the segment associated with said selected one of said segment descriptors in accordance to the level of privilege and the type of access specified in said access information element, wherein each group-type of information is associated with a predetermined ring number indicative of a level of privilege said level of privilege decreasing as the associated ring number increases comprising means for determining the maximum effective address ring number EAR (i.e. minimum level of privilege) of a selected process to access a selected group of information, said means comprising:

(a) first means to store first information indicating the maximum ring number RD (i.e. minimum level of privilege) required to read information from said selected group;

(b) second means to store second information indicating the maximum ring number WR (i.e. minimum level of privilege) required to write information into said selected group;

(c) third means to store third information indicating the maximum ring number MAXR (i.e. minimum level of privilege) required to process information from said selected group; and,

(d) fourth means communicating with said first, second and third means, to determine the maximum of the contents of said first, second and third means whereby the effective address ring number EAR is generated.

The present invention, however, both as to organization and operation thereof may best be understood by reference to the following description which is given by way of example in conjunction with the accompanying drawings in which:

Figure 1 is a block diagram of a computer system utilizing the invention.

Figure 2 is a schematic diagram illustrating the levels of privilege of the invention.

Figure 3 is a flow diagram of the segmented address scheme utilized by the invention.

Figures 4A—4J are schematic diagrams of various novel hardware structures utilized in the invention.

Figure 5 is a schematic diagram of the computer ring protection hardware.

Figure 6 is a schematic diagram of the computer segmented addressing hardware.

Figures 7a—7h and Figures 8a—8c are detailed logic block diagrams of the ring protection hardware.

Figures 9a—9k is a legend of the symbols utilized in the diagrams of the invention.

Figure 10 is a schematic diagram of three stack segments, one each for ring 0, 1 and 3 respectively.

Figure 11A shows the format of the Enter Procedure instruction.

Figure 11B shows the format of a procedure descriptor.

Figure 11C shows the format of a gating procedure descriptor GPD the first word of the segment containing the procedure descriptors.

Figure 11D shows the format of the Exit Procedure instruction.

Figure 12 is a flow diagram of a portion of the Enter Instruction pertaining to ring crossing and ring checking.

Figure 13 schematically shows a segment descriptor and the segment containing procedure descriptors.

Figures 14—16 are flow diagrams showing various operations that are performed when the Enter Procedure instruction is executed.

Figure 17 is a flow chart of the Exit Instruction.

As previously discussed the ring concept of information protection was originated on MULTICS and implemented on various Honeywell (Registered Trade Mark) Computer Systems. The original MULTICS concept required 64 rings or level of privilege and later implementation had the equivalent of two rings on the Honeywell 645 and 8 rings on the Honeywell 6000 (Registered Trade Mark). The embodiment described herein groups data and procedure segments in the system into a hierarchy of 4 rings or classes. (Refer to Figure 2). The 4 rings or privilege levels are identified by integers 0—3; each ring represents a level of privilege in the system with level 0 having the most privilege and level 3 the least. Level 0 is known as the inner ring and level 3 as the outer ring. The basic notion as previously discussed is that a procedure belonging to an inner ring has free access to data in an outer ring. Conversely a procedure in an outer ring cannot access data in an inner ring without incurring a protection violation exception. Transfer of control among procedures is monitored by a protection mechanism such that a procedure execution in an outer ring cannot directly branch to a procedure in an inner ring. This type of control transfer is possible only by execution of a special "procedure-call" instruction. This instruction is protected against misuse in a number of ways. First, a gating mechanism is available to ensure that procedures are entered only at planned entry points called gates when crossing rings. The segment descriptor of such a procedure contains a gate bit indicating that procedures in this segment can be entered only via gates; information regarding these gates is contained at the beginning of the segment and is used by the hardware to cause entry at a legal entry-point. The procedure itself must then verify (in a way which, of necessity depends on the function of the procedure) that it is being legitimately called. A further hardware protection mechanism is available in the case that the calling procedure supplies an address as a parameter; it is then possible that the more privileged procedure would invalidly modify information at this address which the less privileged caller could not have done, since the ring mechanism would have denied him access; an address validation instruction is available to avoid this possibility.

An important convention is required

5
10
15
20
25
30
35
40
45
50
55
60
65

70
75
80
85
90
95
100
105
110
115
120
125
130

here in order to protect the procedure call mechanism. This states that it is not in general permissible to use this mechanism to call a procedure in a less privileged ring and return to the more privileged one. This restriction is necessary since there is no assurance that the procedure in the higher ring will, in fact, return; that it will not, accidentally or maliciously, destroy information that the more privileged procedure is relying upon; or that it will not, accidentally or maliciously, violate the security of the stack (see GLOSSARY for definition). Any of these could lead to unpredictable results and crash the system.

The level of privilege are quite independent of the process control mechanism and there is no notion here of privileged and non-privileged processes as in the IBM system 360 (Registered Trade Mark). Instead the same process can execute procedures at different levels of privilege (rings) subject to the restrictions imposed by the ring mechanism. In this sense the ring mechanism can be viewed as a method for subdividing the total address space assigned to a process according to level of privilege.

The ring mechanism defined herein permits the same segment to belong to up to 3 different rings at the same time i.e. there are 3 ring numbers in each segment descriptor, one for each type of possible access. Thus the same segment can be in ring one with respect to "write" access, ring two with respect to "execute" access and ring three with respect to "read" access. One obvious use for this is in the case of a procedure segment which can be written only by ring zero (perhaps the loader) but can be executed in ring three.

Of the four available rings, two are allocated to the operating system and two to users. Ring zero, the most privileged ring, is restricted to those operating system segments which are critical to the operation of the whole system. These segments form the hard core whose correctness at all times is vital to avoid disaster. Included would be the system information base, those procedures dealing with the organisation of physical memory or the initiation of physical data transfer operations, and the mechanisms which make the system function, like the "exception supervisor, the scheduler, and the resource management".

Ring one contains a much greater volume of operating system segments whose failure would not lead to catastrophe but would allow recovery. Included herein are the language translators, data and message management, and job and process management. Through the availability of two rings for the operating system, the

problem of maintaining system integrity is made more tractable, since the smaller hard core which is critical is isolated and can be most carefully protected.

Rings two and three are available to the user to assign according to his requirement. Two important possibilities are debugging and proprietary packages. Programs being debugged may be assigned to ring two while checked out programs and data with which they work may be in ring two; in this way the effect of errors may be localized. Proprietary programs may be protected from their users by being placed in ring two while the latter occupy ring three. In these and other ways, these two rings may be flexibly used in applications.

The General Rules of the Ring System

1. A procedure in an inner ring such as ring 2 on Figure 2 has free access to data in an outer ring such as ring 3 and a legal access (arrow 201) results. Conversely a procedure in an outer ring such as ring 3 cannot access data in an inner ring such as ring 2 and an attempt to do so results in an illegal access (arrow 202).

2. A procedure in an outer ring such as ring 3 can branch to an inner ring such as ring 1 via gate 204 which results in a legal branch 203, but a procedure operating in an inner ring such as ring 2 may not branch to an outer ring such as ring 3.

3. Each segment containing data is assigned 2 ring values, one for read (RD) and one for write (WR). These ring values specify the maximum ring value in which a procedure may execute when accessing the data in either the read or write mode.

Each time a procedure instruction is executed, the procedure's ring number (effective address ring, EAR) is checked against the ring numbers assigned to the segment containing the referenced data. The EAR is the maximum number of process ring numbers in the processor instruction counter (see later description) and all ring numbers in base registers and data descriptors found in the addressing path. Access to the data is granted or denied based on a comparison of the ring numbers. For example, if a system table exists in a segment having a maximum read/ring value of 3 and a maximum write/ring value of 1, then a user procedure executing in ring 3 may read the table but may not update the table by writing therein.

Procedure Calls and the Stack Mechanism:

The procedure call and stack mechanism is an apparatus being described herein Procedure calls are used to pass from one procedure to another; to allow user procedures to employ operating system services: and to achieve a modular

structure within the operating system. A procedure call is effected by instructions and a hardware recognized entity called a stack.

5 A stack is a mechanism that accepts, stores and allows retrieval of data on a last-in-first-out basis. Stacks reside in special segments called stack segments. A stack segment consists of a number of contiguous parts called stack frames which are dynamically allocated to each procedure. 10 The first stack frame is loaded into the low end of the segment and succeeding frames are loaded after it. The last frame loaded is considered the top of the stack. A T-register 114 (see Figure 1) locates the top of the stack for the currently active process. A virtual T-register exists in the process control block (PCB) of all other processes 20 in the system.

A stack frame consists of three areas: a work area in which to store variables, a save area in which to save the contents of registers, and a communications area in which to pass parameters between procedures. 25 Prior to a procedure call, the user must specify those registers he wishes saved and he must load into the communications area the parameters to be passed to the called procedure. When the call is made, the hardware saves the contents of the instruction counter and specified base registers to facilitate a return from the called procedure. 30

Each procedure call creates a stack frame within a stack segment and subsequent calls create additional frames. Each exit from one of these called procedures causes a stack frame to be deleted from the stack. Thus, a history of calls is maintained which facilitates orderly returns. 35

To ensure protection between procedures executing in different rings, different stack segments are used. There is one stack segment corresponding to each protection ring per process. A process control block (PCB) contains three stack base words (SBW) which point to the start of the stack segment for rings 0, 1 and 2 associated with the process. The ring 3 stack segment can never be entered by an inward call; therefore, its stack starting address is not required in the PCB. 40

The procedure call is used by users who have written their programs in a modular way to pass from one program module to another. It is used by user programs to avail themselves of operating system services. It is used by the operating system itself to achieve a responsive modular structure. The procedure call as is described in the above referenced patent application is effected by hardware instructions and the hardware recognizable stack mechanism. 45 50 55 60 65

The main requirements on a procedure call mechanism are:

- 1. Check the caller's right to call the caller;
- 2. Save the status of the caller which includes saving registers, instruction counter (for return), and other status bits;
- 3. Allow for the passing of parameters;
- 4. Determine valid entry point for the called procedure;
- 5. Make any necessary adjustments in the addressing mechanism;
- 6. Enter the new procedure.

When the called procedure terminates or exits, whatever was done in the call must be undone so that the status of the calling procedure is restored to what it was before the call. 70 75 80

As a preliminary to making a procedure call, the instruction PREPARE STACK is executed. This instruction causes those registers specified by the programmer in the instruction to be saved in the stack. It causes the status register (see Figure 1) to be saved, and provides the programmer with a pointer to parameter space which he may now load with information to be passed to the called procedure. 85 90

Another instruction ENTER PROCEDURE permits the procedure call via the following steps corresponding to the requirement specified above: 95

- 1. Ring checking—the caller's ring is checked to make sure that this ring may call the new procedure; the call must be to a smaller or equal ring number; and if ring crossing does occur the new procedure must be gated through a gate 204 of Figure 2. The new ring number will then be that of the called procedure.
- 2. The instruction counter is saved;
- 3. Base register 0 (see Figure 1) is made to point effectively to the parameters being passed;
- 4. The entry-point of the called procedure is obtained from a procedure descriptor whose address is contained in the ENTER PROCEDURE INSTRUCTION;
- 5. A point to linkage information is loaded in base register number 7.
- 6. The new procedure is entered by loading the new ring number and the address of the entry-point in the instruction counter.

The remainder of the current stack-frame is also available to the called procedure for storage of local variables. When the called procedure wishes to return, it executes the instruction EXIT PROCEDURE. The registers and the instruction counter are then restored from their saving areas in the stack. Referring to Figure 1 there is shown a block diagram and a computer hardware 100 105 110 115 120

125

130

system utilizing the invention. A main memory 101 is comprised of four modules of metal-oxide semi-conductor (MOS) memory. The four memory modules 1-4 are interfaced to the central processor unit 100 via the main store sequencer 102. The four main memory modules 1-4 are also interfaced to the peripheral subsystem such as magnetic tape units and disk drive units (not shown) via the main store sequencer 102 and the 10C (not shown). The main store sequencer gives the capability of providing access to and control of all four memory modules.

Operations of the CPU are controlled by a read only memory ROM, herein called the control store unit 110.

The control store interface adapter 109 communicates with the control store unit 110, the data management unit 106, the address control unit 107 and the arithmetic logic unit 112 for directing the operation of the control store memory. The control store interface adapter 109 includes logic for control store address modification, testing, error checking, and hardware address generation. Hardware address generation is utilized generally for developing the starting address of error sequencers or for the initialization sequence.

The buffer store memory 104 is utilized to store the most frequently used or most recently used information that is being processed by the CPU.

The data management unit 106 provides the interface between the CPU 100 and main memory 101 and/or buffer store memory 104. During a memory read operation, information may be retrieved from main memory or buffer store memory. It is the responsibility of the data management unit to recognize which unit contains the information and strobe the information into the CPU registers at the proper time. The data management unit also performs the masking during partial write operations.

The instruction fetch unit 108 which interfaces with the data management unit 106, the address control unit 107, the arithmetic and logic unit 112 and the control store unit 110 is responsible for keeping the CPU 100 supplied with instructions.

The address control unit 107 communicates with the instruction fetch unit 108, the buffer store directory 105, the main store sequencer 102, the arithmetic logic unit 112, the data management unit 106, and the control store unit 110 via the control store interface adapter 109. The address control unit 107 is responsible for all address development in the CPU.

Interfacing with the address control unit

107, the instruction fetch unit 108 and the control store unit 110 is the arithmetic logic unit 112 which is the primary work area of the CPU 100. Its primary function is to perform the arithmetic operations and data manipulations required of the CPU.

Associated with the arithmetic logic unit 112 and the control store unit 110 is the local store unit 111 which typically is comprised of a 256-location (32 bits per location) solid state memory and the selection and read/write logic for the memory. The local store memory 111 is used to store CPU control information and maintain ability information. In addition, the local store memory 111 contains working locations which are primarily used for temporary storage of operands and partial results during data manipulation.

The central processing unit 100 typically contains 8 base registers (BR) 116 which are used in the process of address computation to define a segment number, an offset, and a ring number. The offset is a pointer within the segment and the ring number is used in the address validity calculation to determine access rights for a particular reference to a segment.

The instruction counter 118 communicates with the main memory local register (MLR) 103 and with the instruction fetch unit 108, and is a 32-bit register which contains the address of the next instruction, and the current ring number of the process (PRN). Also contained in the central processing unit is a T register 114 which also interfaces with the instruction fetch unit 108 and is typically a 32-bit register containing a segment number and a 16-bit or 22-bit positive integer defining the relative address of the top of the procedure stack. The status register 115 is an 8-bit register in the CPU which among other things contains the last ring number—i.e. the previous value of the process ring number (PRN).

The main memory 101 is addressed by the memory address register (MAR) 119, and the information addressed by (MAR) 119 is fetched and temporarily stored in the memory local register (MLR) 103.

Referring now to Figure 3 there is shown a flow diagram of the general rules for segmented address development shown in detail in the above mentioned copending patent application No. 21630/74, Serial No. 1,465,344. Figure 3 when read in conjunction with the above referenced patent application is self-explanatory. There is however one major difference between the address development as shown on Figure 3 to that of the above mentioned application and that is that in the address development of Figure 3 of the instant application as many as 16 levels of

5

10

15

20

25

30

35

40

45

50

55

60

65

70

75

80

85

90

95

100

105

110

115

120

125

130

indirection may be utilized in the address development whereas in the above referenced application the levels of indirection were limited to a maximum of two. This of course is a matter of choice with the designer and in no way alters the high level inventive concept.

Referring now to Figures 4A—4J, Figures 4A and 4B show the format of the instruction counter designated by reference numeral 118 on Figure 1. The instruction counter (IC) 118 is a 32-bit register which contains the address of the next instruction, and the current ring number of the process (PRN). Referring specifically to Figures 4A and 4B the TAG is a 2-bit field which corresponds to the TAG field of data descriptors shown and described in the above reference application entitled "Segmented Address Development". PRN is a 2-bit field which defines the current ring number of the process to be used in determination of access rights to main storage. SEG is typically either a 12-bit or a 6-bit field which defines the segment number where instructions are being executed. The OFFSET is typically either a 16-bit or a 22-bit field which defines the address of the instruction within the segment SEG.

Figures 4C—4F show the format of segment descriptors with Figures 4C and 4D showing the first and second word of a direct segment descriptor whereas Figures 4E and 4F show the first and second word of an indirect segment descriptor. Segment descriptors are two words long each word comprised of 32 bits. Referring to Figures 4C—4D which show the first and second word respectively of a direct segment descriptor, P is a presence bit. If P equals one, the segment defined by the segment descriptor is present in main storage. If P equals zero, the segment is not present and a reference to the segment descriptor causes a missing segment exception. All other fields in a segment descriptor have meaning only if P equals one. A is the availability bit. If A equals zero, the segment is unavailable (or locked) and a reference to the segment causes an unavailable segment exception. If A equals one, the segment is available (or unlocked, and can be accessed). I is the indirection bit. If I equals zero, the segment descriptor is direct. If I equals one, the segment descriptor is indirect. U is the used bit. If U equals zero, the segment has not been accessed. If U equals one, the segment has been accessed. U is set equal to one by any segment access. W is the written bit. If W equals zero, no write operation has been performed on the segment. If W equals one, a WRITE operation has been performed on the segment. W is set to one by any WRITE

operation. GS is the gating-semaphore bits. When the procedure call mechanism referred to above requires that the segment be a gating segment or when the process communication mechanism (not shown) requires that the segment be a segment descriptor segment (SD) the GS bits are examined. To be a valid gating segment, the GS bits must have the value 10. To be a valid SD segment, the GS bits must have the value 01. If a gating or SD segment is not required, these bits are ignored. The BASE is a 24-bit field which defines the absolute address in quadruple words of the first byte of the segment. This field is multiplied by 16 to compute the byte address of the segment base. The SIZE is a field which is used to compute the segment size. If the segment table number, subsequently referred to as STN, is greater or equal to zero but less than or equal to six, the SIZE field is 18 bits long. The STN is a field indicating the segment table entry STE for selecting a segment descriptor. If the STN is greater than or equal to 8 but less than or equal to 15, the SIZE field is 12 bits long. The number of bytes in the segment is equal to 16 times (SIZE+1). If SIZE equals zero, the segment size is 16 bytes. RD is the read access field. This is a 2-bit field which specifies the maximum EAR (effective address ring number) for which a read operation is permitted on the segment. (A procedure is always permitted to read its own segment if EAR equals PRN). WR is the write access field. This is a 2-bit field which specifies the maximum EAR for which a write operation is permitted on the segment and the minimum PRN at which the segment may be executed. MAXR is the maximum ring number. This is a 2-bit field which specifies the maximum PRN at which the segment may be executed. WP is the write permission bit. This bit indicates whether a WRITE operation may be performed on the segment. If WP equals zero, no WRITE operation may be performed. If WP equals one, a WRITE operation may be performed if EAR is greater than or equal to zero but less than or equal to WR. EP is the execute permission bit. This bit specifies whether the segment may be executed. If EP equals zero, the segment may not be executed. If EP equals one, the segment may be executed at any PRN for which PRN is greater than or equal to WR but less than or equal to MAXR. MBZ is a special field which must be set to zero by software when the field is created, before its initial use by hardware.

Referring to Figures 4E—4F the definitions of the various fields are similar as above however word 0 includes a LOCATION field and word 1 includes a

5
10
15
20
25
30
35
40
45
50
55
60
65

70
75
80
85
90
95
100
105
110
115
120
125
130

RSU field. The LOCATION field is a 28-bit field which defines the absolute address of a direct segment descriptor. The value in the LOCATION field must be a multiple of 8. The RSU field is a special field which is reserved for software use.

Figures 4G—4H show the format of the base registers (BR) which are used in the process of address computation to define a segment table number, a segment table entry number, an offset, and a ring number. There are typically 8 base registers as shown by reference numeral 116 on Figure 1. A base register is specified or identified as base register 0 through 7. The size of a base register is 32 bits long. The base register format of Figure 4G is utilized for small segment i.e. where STN is greater or equal to 8 but less than or equal to 15, whereas the format of base register of Figure 4H is utilized for large segments i.e. STN is greater or equal to zero but less than or equal to six. Referring to Figures 4G—4H, TAG is a 2-bit field which corresponds to the TAG of a data descriptor referenced previously. RING is a 2-bit field which contains the ring number associated with the segmented address for protection purposes. SEG is a field previously referred to, which identifies a segment described in a segment table. STN is the segment table number, and STE is the segment table entry number. OFFSET is a 16-bit field or a 22-bit field depending on segment table number, which defines a positive integer. The OFFSET is used in the process of address development as a pointer within a segment.

Referring to Figures 4I—4J there is shown the format of the T-register. The T-register is a 32-bit register containing a segment number and a 16-bit or 22-bit positive integer defining the relative address of the top of the procedure stack previously mentioned. The T-register is shown by reference numeral 114 on Figure 1. The various fields of the T-register have the same definition as described above.

Referring now to Figures 3 and 4A—4J a more defined description of absolute address calculation and access checking is made. In general absolute address calculation consists of fetching a segment descriptor specified by STN and STE and using the segment descriptors in four ways: access checking, computation of the absolute address, bound checking, and updating (U and W flags). As described in copending patent application No. 21630/74, (Serial No. 1,465,344) the absolute address may be direct or indirect and is derived by first deriving an effective address from STN, STE, and SRA (segment relative address). STN is extracted from bits 4 through 8 of the base register BR specified

in the address syllable of an instruction. If STN is 7, an out of segment table word array exception is generated. STE is extracted from the base register specified in the address syllable. If STN 4:4 (i.e., beginning at bit 4 and including the next 4 bits) is greater than or equal to zero or less than or equal to six, STE is in a base register bits 8 and 9. If STN 4:4 (i.e. 4 bits beginning at bit 4) is greater than or equal to 8 but less than or equal to 15, STE is in a base register BR bits 8 through 15. The segment relative address SRA for direct addressing is computed by adding the displacement in the address syllable; the offset of the base register BR; and the 32-bit contents of an index register, if specified in the address syllable. The sum of these three quantities is a 32-bit unsigned binary integer which must be less than the segment size appropriate to the segment STN, STE.

Indirect addressing is developed by fetching a data descriptor and developing an address from that descriptor. The effective address of the data descriptor is computed as in the direct addressing case with the exception that the index register contents are not used. In developing the address from the data descriptor the effective address may be computed by an indirection to segment ITS descriptor and an indirection to base ITBB descriptor. If the descriptor is ITS the STN and STE are extracted from the descriptor in the same manner as from a base register. SRA is computed by adding the displacement in the descriptor and the contents of an index register as specified in the syllable. If the descriptor is an ITBB descriptor then STN and STE are extracted from the base register specified in the BBR field (i.e. the base register implied by ITBB descriptor) of the descriptor as in direct addressing. SRA is computed by adding the displacement in the descriptor, the offset of the base register, and the contents of an index register is specified in the address syllable.

As shown on Figure 3 the indirection process may be extended up to 16 levels.

Every effective address contains protection information which is computed in address development and checks for access rights by the ring protection hardware of the absolute address calculation mechanism. The effective address contains protection information in the form of an effective address ring number EAR (see Figures 2J and 2K of above application No. 21630/74, (Serial No. 1,465,344). The EAR is computed from the base register ring number BRN and from the current process ring number PRN by taking the maximum ring number. In developing the EAR for indirect addressing

a somewhat more tedious but essentially similar procedure as indirect addressing is used. In indirect addressing the EAR for extraction of the first descriptor (EAR 1) is once again the maximum of the ring number from the base register specified in the address syllable and the current process ring number PRN in the instruction counter 115 of Figure 1 and stored in 00 register 512 of Figure 5. The EAR for extraction of the second descriptor (EAR 2), of multiple level indirection is the maximum of:

- a. EAR 1;
- b. The ring number in the first descriptor if indirection is indirection to segment;
- c. The ring number from a base register 116 utilized as a data base register BBR if the first descriptor is an indirection to segment descriptor ITBB.

The EAR for extraction of the data of multiple level indirection is the maximum of:

- a. EAR 2;
- b. The ring number in the second descriptor if it is an indirection segment descriptor ITS;
- c. The ring number in one of the base registers utilized as a data base register BBR if the second descriptor is an indirection to base descriptor ITBB.

Referring now to Figures 5 and 6, the transfers and manipulation of the various type ring numbers will be described at the system level. Detailed logic block diagrams for effecting the transfers and operations of Figure 5 will be later described. Referring first to Figure 6 an associative memory 600 is utilized in segmented address development. The associative memory 600 comprises essentially a UAS associator 609 which has circuitry which includes associative memory cells, bit sense amplifiers and drivers, and word sense amplifiers and drivers (not shown). A word or any part of a word contained in UAS associator 609 may be read, compared to another word with a match or no match signal generated thereby, or be written either in whole or in a selected part of the associator 609. For example, US register 607 may contain a segment number which may also be in the associative memory 600. A comparison is made with UAS associator 609 and if a match is found a "hit" results. The match or "hit" signal is provided to encoder 610. The function of encoder 610 is to transform the "hit" signal on one of the match lines to a 4 bit address. Encoder 610 provides this 4 bit address to UAB associator buffer 611 so that the information contained in that particular location of UAB associator buffer 611 is selected. Information in UAB associator buffer 611 may be transferred to UV register 613 for temporary storage or

for transfer to QA or QB bus 614 and 615 respectively. By thus locating a prestored segment number of the associative memory 600 (which may have been placed there after a generation of an absolute address) regeneration of the same address is not necessary. In the drawing of Figure 6, UAB associator buffer 611 is shown as storing a first and second word of a segment descriptor; however other types of information may just as well be stored therein. This buffer 611 provides a function similar to that of buffer 104 in the more generalised diagram of Figure 1.

As mentioned supra the development of an absolute address of an operand from an effective address is disclosed in patent application No. 21630/74, (Serial No. 1,465,344). Briefly and with reference to Figure 6 any of 8 base registers 602 are addressed via UG and UH registers 603 and 604 respectively which contain base register addresses from an instruction address syllable or base register specified by the instruction formats. The base register 602 contain such information as TAG, base register ring number BRN, segment table number STN, segment table entry STE and OFFSET as shown or contained by base registers 1 and 2 of the group of base registers 602. Writing into the base registers is performed under micro-op control by UWB logic 601. For example it is shown that information from the UM register 502 of Figure 5 may be written into bit positions (2, 3) of a selected base register; also information from the QA bus may be written into the base registers and provisions are made to clear a selected base register i.e. write all zeroes. Reading out of any of the base registers is performed by UBR logic 605. In general the UBR logic 605 permits the appropriate base register to be strobed out onto bus QA or QB, or into UN register 608. Note that UN register 608 holds bits 8 through 31 of the base registers which is the OFFSET part of the segmented address. Moreover UBR logic 605 when addressed by an address contained in instruction buffer IB (not shown) reads out the segment number SEG (which is comprised of STN and STE) into US register 607 via UBS transfer logic 606. The comparison of the segment number SEG in US register 607 with the associative memory 600 may then be performed as previously described. It will be noted that bits (4—15) of QA bus 614 may also be read into or from US register 607. Similarly bits (8—31) from QA bus 614 may read into UN register 608. Also bits (9—11) of the US register 607 may be read into QA bus 614 as denoted by US (9—11) arrow (the arrows into various register and/or logic circuitry denote the source of data and that followed

70

75

80

85

90

95

100

105

110

115

120

125

130

by a number denote the bit numbers of that data).

Referring now to Figures 5 and 6, a 2-bit UP register 501 stores the current process ring number PRN. The current process ring numbers PRN is obtained from bits 2 and 3 of the instruction counter (118 or Figure 1) via bits IC (2—3) of the QA bus 614 of Figure 6. Bits IC (2—3) of QA bus 614 are transferred to 2-bit UV register 503 under control of a micro-operation UV9QA0. The micro-operations are obtained from micro-instructions in the control store unit 110. (On Figure 5 the dot surrounded by a circle indicates a micro-operation and the first two letters of the name of the micro-operation indicate the destination of the data to be transferred; the fourth and fifth letters indicate the source of the data transferred; the third character indicates whether a full or partial transfer is made with F indicating a full transfer while the sixth character indicates whether the signal doing the transferring is high or low with even numbers indicating a low signal and odd numbers indicating a high signal. As an example of the use of this convention bits 2 and 3 on QA bus indicating the tail of the arrow QA (2, 3) indicate PRN is the PRN process ring number that is being transferred under control of the micro-op UV9QA0 which says the transfer is made to register UV, is a partial transfer of the bus QA, and the source of the data is the bus QA and is an unconditional transfer as indicated by the sixth character being 0. Transfer to UV register from QA bus source is unconditional. This 0 will be the corresponding seventh character in the logic file name of the subcommand UV9QA1φ. Once the process ring number PRN is transferred from the QA bus 614 to the UV register 503 another transfer takes place under control of the micro-operation UM9UV0 from UV register 503 to UM register 502. Finally another transfer takes place from UM register 502 to UP register 501 under control of a micro-operation UP9UM0.

Two bit register UM 502 is utilized to generate the effective address ring number EAR during ITS and ITBB (i.e. indirection to segment and indirection to base), (EAR=MAX (BRN, PRN, DRN, BBR (BRN) etc.) address formation for address syllable 1 and address syllable 2 type instruction format. The EAR is generated according to the rules previously enunciated by utilizing one or more tests shown in block 510 and the maximum of the ring number is obtained and stored in UM register 502 which stores the effective address ring number EAR (detailed logic or making the comparisons of block 510 are later shown and described in detail). The

UO register is used to save address syllable 1 effective address ring number EAR in the event the address syllable 2 is being utilized to extract EAR 2.

Two-bit UV register 503, and 2-bit UW register 504 is utilized mainly as storage for various ring numbers that are obtained from the outside of the ring checking hardware of Figure 5 and transferred or processed to other parts of the ring checking hardware. For example the base register ring number BRN is transferred from bit positions 2 and 3 of UBS transfer logic 606 to UV register 503 under control of the micro-operation UVFBS0; the maximum ring number MAXR of word 2 of the segment descriptor (also shown stored in bits 36 and 37 of UAB associator buffer 611) is transferred from UAB buffer 611 to UV register 503 under control of the micro-operation UVFAB1; also bits 34 and 35 of UAB buffer 611 which is the write ring number WR is transferred to UV register 503 under control of micro-operation UVFAB0. UW register 504 has similar transfers of other ring numbers from various parts of the system. For example bits 34 and 35 which are the write ring number WR of UAB buffer 611 may also be transferred to UW register 504 under control of micro-operation UWFAB1; bits 32 and 33, the read RD ring number of UAB buffer 611 may also be transferred to UW register 504 under control of micro-op UWFAB0; also bits 0 and 1 of QA bus 614 may be transferred to UW register 504 under control of micro-operation UW9QA0. Note also several transfer paths of UW register 504 into UV register 503 under control of the micro-operation UV9UW0; the transfer path of UV register 503 into UM register 502 under control of micro-operation UM9UV0; the transfer path of UM register 502 into UP register 501 under control of the micro-operation UP9UM0; the transfer path of UP register 501 into UM register 502 under control of micro-operation UM9UP0; the transfer path of UM register 502 into UO register 512 under control of micro-operation UO9UM0; and finally the transfer path of UO register 512 into UM register 502 under control of the micro-operation UM9UO0.

Briefly therefore UP register 501 holds the current process ring number PRN; UM register 502 and UO register 512 are utilized for transfer operations and also to generate the EAR; UV register 503 may shore for various purposes and at different times the current process ring number PRN, the base register ring number BRN, the maximum ring number MAXR, the write ring number WR, or the read ring number RD. UW register 504 may at various times hold the read ring number RD, the write ring

number WR, and bits 0 and 1 of bus QA. UMR 505 is logic, the details of which are shown on Figure 8d, which compares the contents of registers UM and UV and produces the greater of the two values in the registers and this value is stored in UM register 502 under micro-operation control UMFMR0. This is one way of generating the effective address ring number EAR. UMR logic 505 may also produce the greater value of the contents of register UP or of bits 2 and 3 of UBS logic 606. This is another method and/or additional step in generating the effective address ring number EAR. UMR logic 505 is also utilized to determine whether or not a write violation has occurred by transferring a write ring number WR into UV register 503 and then comparing the contents of the UM register 502 (holding EAR) with the contents of UV register 503 in order to determine which one has the greater contents. Since UM register 502 stores the effective address ring number EAR a comparison of the UM register and the UV register will indicate whether EAR is greater than WR or vice versa. If WP (i.e. write permission bit in the segment descriptor) is equal to 1 and if EAR lies in the range of $0 \leq EAR \leq WR$ then a write operation may be performed into the segment. Note that UMR logic 505 may have inputs directly or indirectly from all registers 501—504, from other logic 506, 507 and also from UBS logic 606.

UWV logic 506 corresponds to the detail logic of Figure 8a. UWV logic 506 has inputs directly or indirectly from registers 501—504 and from logic 505, 507 respectively and generates an execute violation signal when a comparison of UW, UM and UV registers 504, 502, and 503 respectively indicates that the statements that the maximum ring number MAXR is greater or equal to the effective address ring number EAR, and that EAR is greater or equal to the write ring number WR are not true i.e. in order for a procedure to be able to execute in a given segment indicated by the effective address the maximum ring number MAXR must be greater or equal to the effective address ring number EAR must be equal or greater than the write ring number WR. UWV logic 506 also performs tests shown in block 510. Indications may be given that the contents of UW register is less than or equal to the contents of the UV register; the contents of the UM register is greater than or equal to the contents of the UV register; the contents of the UV register is equal to the contents of the UM register; the contents of the UV register is greater or equal to the contents of the UM register; and the

contents of the UM register is greater than the contents of the UW register. Of course when performing these tests different values of ring numbers may occupy the registers.

UEP logic 507 corresponds to the detail logic of Figure 8b. UEP logic 507 in combination with UWV logic 506 generates the read violation exception. However the read violation exception may be overridden if the effective address ring number EAR equals the current process ring number PRN, since a procedure is always permitted to read its own segment, and if the segment number of the procedure segment descriptor (not shown herein) and the segment number of the address syllable utilized in generation of the effective address are the same.

To illustrate the overriding of the read violation signal assume that the effective address read number EAR is greater than the read number RD which would generate a read violation high signal which would be applied as one input of AND gate 522. However the read violation exception signal may not be generated even though there is a read violation signal if the following two conditions exist:

1. The effective address ring number EAR is equal to the process ring number PRN; i.e. the contents of register UM is equal to the contents of the register UP; and,

2. The segment number contained in the address syllable of the segment in which a procedure desires to read is equal to the segment number of the procedure segment descriptor (not shown) of the current procedure in execution and this is indicated by setting a bit called a P bit and located as the thirteenth bit of UE register 650. (UE register 650 is a store for the contents of UAS associator 609 when a "hit" has resulted by a comparison of the contents of US register 607). Since this example assumes that EAR equals PRN, UEP logic 507 will apply a high signal to AND gate 520 as one input, and since it is also assumed that the segment number SEG of the address syllable of the segment being addressed is equal to the segment number SEG of the procedure segment descriptor (not shown) of the currently executing procedure, then the P bit of the procedure segment descriptor will be set and hence the other input applied to AND gate 520 will be high thus enabling AND gate 520; a high signal is therefore applied to the input of inverter 521 resulting in a low signal at the output of inverter 521 which low signal is then applied as another input of AND gate 522. Since there is a low signal to AND gate 522 no read violation exception signal can be generated by amplifier 523 even if

5
10
15
20
25
30
35
40
45
50
55
60
65

70
75
80
85
90
95
100
105
110
115
120
125
130

the third input signal applied to AND gate 522 is high.

To illustrate how a read violation signal is generated and not overridden, assume that the output of UEP logic 507 indicates that the contents of UM register is not equal to the contents of UP register. Then that input to AND gate 520 would be low and hence AND gate 520 would not be enabled and its output would be low and would be applied to the input of inverter 521. Since the input of inverter 521 is low its output would be high which would be applied as one input of AND gate 522. If also the effective address ring number EAR is greater than the read ring number RD (i.e. contents of UM register is greater than contents of UW register) that signal would be high and would be also applied to another input of AND gate 522. AND gate 522 has still a third input which must also be high in order to enable AND gate 522. This third input is high when AND gate 526 is enabled. Since AND gate 526 has one input terminal which is high when the 00 terminal of URVIF flop 524 is low, AND gate 526 is enabled by applying the micro-operation read violation interrogate signal AJERVA to one input terminal of AND gate 526 while the 00 terminal of URVIF flop 524 is low. Thus AND gate 522 will have all input terminals high, generating the read violation exception signal.

The execute violation exception is generated in two ways. It was seen earlier that an execute violation signal results when UWV logic 506 indicates that the inequalities $WR < EAR$, and $EAR \leq MAXR$ are not true. This high execute violation signal is applied to a one-legged AND gate 550 which in turn is applied to the input terminal of two-legged AND gate 553 via amplifier 552. When an execute violation interrogate micro-operation signal AJEEVA is applied as another input of two-legged AND gate 553, this gate is enabled which in turn generates the execute violation exception via amplifier 554. The other method by which the execute violation exception is generated by the execute violation hardware 511 is when the execute permission bit EP is not set. When this condition is true it is indicated by the seventh bit of UY register 613 being high; this bit is then applied to the input terminal of one-legged AND gate 551 which is applied as a high signal to one input terminal of AND gate 553 via amplifier 552. When the execute violation interrogate micro-operation signal AJEEVA goes high, AND gate 553 is enabled and generates an execute violation exception via amplifier 554.

The write violation exception is also

generated in two ways. It was seen previously how the UMR logic 505 generates a write violation signal when $EAR > WR$. This write violation signal is applied to one input terminal of AND gate 545. AND gate 545 is enabled when its second input terminal goes high thus generating a write violation exception through amplifier 547. The second input terminal of AND gate 545 goes high when AND gate 542 is enabled. AND gate 542 is enabled when the input signals applied to its input terminals are high. One input signal is high when UWVIF flop 541 is low which in turn applies a low signal to the input terminal of inverter 543 which in turn applies a high signal to one input terminal of AND gate 542; the other input signal is high when the write violation interrogate micro-op signal AJEWVA is high and this happens when it is desired to interrogate a procedure for the write violation exception. (Flip-flops URVIF, URNIF, and UWVIF are set low when any interrupts or software occurs). (UWV2F, URV2F, and URN2F flip-flops are utilized to store back-up excess checking information for ring checking). The other method for generating a write violation exception is when the write permission bit WP is not set. This condition is indicated by bit 6 of UV register 613 being high. When this condition exists and the high signal (i.e. the sixth bit of UV register) is applied as one input of AND gate 546 and the interrogate signal

AJEWVA is high and applied as another input of AND gate 546, then AND gate 546 is enabled and a write violation exception occurs via amplifier 547.

Logic circuitry 591 comprised of flip-flops 532 and 533 in conjunction with amplifier 530 and AND gate 531 and inverter 530A permit the formation in register UM 502 of the maximum value of ring number (i.e. EAR) under control of a splatter instruction subcommand (not described herein) from the instruction fetch unit IFU. Assuming URN1F flip-flop 532 is set to logical 0 whereas URN2F flip-flop 533 is set to logical 1, then during the execution of the splatter subcommand, input terminal 531A of AND gate 531 will be high; therefore if flip-flop 532 is low (logical 0) then the signal will be inverted by inverter 530A and AND gate 531 will be enabled. Hence the maximum value of the contents of UP register 501 or bits 2 and 3 of logic vector UBS 606 will be strobed into UM register 502. Conversely if flip-flop 532 is a logical 1, then the contents of UM register 502 is not changed via the above mentioned sources and the EAR derived in UM register 502 via the addressing process of indirection is the one utilized. Flip-flop

533 is the back-up store for the EAR of address-syllable 2 when utilized.

Referring now to Figures 7 and 8 and Figure 5 there is a correspondence wherein the detailed logic for hardware in Figure 5 is shown in Figures 7 and 8 as follows: Figure 7a and UW register 504; Figure 7b and UV register 503; Figure 7c and block 590; Figure 7d and block 591; Figure 7e and block 592; Figure 7f and UP register 501; Figure 7g and UO register 512; Figure 7h and UM register 502; Figure 8a and UWV logic 506; Figure 8b and UEP logic 507; and Figure 8d and UMR logic 505.

Referring to Figure 7a, the UW register 504 is comprised of two flip-flops 715a and 720a respectively, each flip-flop capable of holding one bit of information of the UW register. Coupled to flip-flop 715a are 4 AND gates 711a-714a which are OR'ed together, with each gate (except gate 713a) having two input terminals, and with at least one signal applied to each input terminal. AND gate 714a has one of its input terminals coupled to the set terminal OW00010 of the flip-flop 715a. Flip-flop 715a is also coupled to the terminal H27 for receiving from a clock a timing signal called a PDA signal. Flip-flop 720a coupled to AND gates 716a-719a which are OR'ed together. One input terminal of AND gate 716a is coupled to an input terminal of AND gate 711a; one input terminal of AND gate 717a is coupled to one input terminal of AND gate 712a and one input terminal of AND gate 719a is coupled to an input terminal of AND gate 714a, whereas the other input terminal of AND gate 719a is coupled to the set terminal UW00110 of the flip-flop 720a. Flip-flop 720a is also coupled to the H27 terminal for receiving PDA pulses.

AND gates 701a-704a are OR'ed together each having their output terminal coupled to the input terminal of inverter 705a. AND gate 706a is coupled to amplifier 708a; whereas AND gate 707a is coupled to amplifier 709a; one input terminal of AND gate 706a is coupled to one input terminal of AND gate 707a. The output terminal of inverter 705a is coupled to one input terminal of AND gate 714a and 719a; the output terminal of amplifier 708a is coupled to the input terminal of AND gate 713a and the output terminal of amplifier 709a is coupled to the input terminal of AND gate 718a.

The signals applied to the inputs of AND gates and the signals derived as outputs from amplifier, inverters, or flip-flops are designated by letters forming a special code. Since both data signals and control signals are either applied or derived there are two codes, one code for the control signals and one code for the data signals.

The code for the control signals are previously described in detail and is summarized here. Briefly the first two characters of a control signal indicate the destination of data to be transferred; the third character indicates whether a full or partial transfer is to be effected with the letter F indicating full transfer and any other character indicating a partial transfer; the fourth and fifth character indicates the source of the data, and if the source is identified by more than two letters only the last two letters need be used; the sixth and seventh characters are usually numerals and indicate whether the signal is high or low i.e. an odd numeral in the sixth position indicates assertion and an even numeral in the sixth position indicates negation; the seventh position indicates whether this is the first, second, third, etc. level of occurrence of the signal. Data, on the other hand, is indicated differently. The first three characters of data indicates the source of the data, the fourth and fifth characters which may be numerals indicate the bit positions where the data is located in the source, and the sixth and seventh position are similar to the control signals in that they indicate whether the signal is high or low and the level of occurrence of the signal. Generally the format itself indicates whether the signal is a control signal or a data signal and by reference to Figures 5 and 6 the source and destination may be determined. There are exceptions to this general rule and they will be spelled out in the specification, and addendum.

As an example of this convention it will be noted on Figure 7a that the following signals are control signals: UWFAB11, UWFAB10, UW9QA10. The following signals are data signals UAB3410, UAB3210, UAB3510, UAB3310, QA00110, and QA00010. The following signals are exception PDARG10 is a timing signal whose source is the PDA clock; UWHOL10 is a hold signal for holding the information in the flip-flops 715a and 720a UWOBK10 and UW1BK10 are back-up logic whose main function is to extend the input capability of flip-flops 715a and 720a by connecting the UW register which is in fact formed by flip-flops 715a and 720a, to bit zero and bit 1 represented by flip-flops 715a and 720a respectively; and finally USCLR10 is the clear signal for clearing and setting the flip-flops to zero.

As an illustration of the above mentioned convention herein adopted the signal UWFAB11 applied to the input of one-legged AND gate 702a is a control signal which transfers data (bits 34 and 35) contained in UAB associator buffer 611 (the U in the signal has been omitted) to UW register 504 and is a full transfer to the

5
10
15
20
25
30
35
40
45
50
55
60
65

70
75
80
85
90
95
100
105
110
115
120
125
130

UW register 1; the odd number indicates the signal is assertion. Signal UWFAB10 applied to the input of one-legged AND gate 703a is a control signal with the same source and destination as the signal applied to AND gate 702a except that bits 32 and 33 of UAB are transferred to UW register. The signal UW9QA10 applied to one-legged AND gate 704a is also a control signal wherein data is transferred from QA bus 614 to the UW register and may be a partial transfer. The signal QA00010 applied to AND gate 706a is a data signal where data is on QA bus 614 (the third position is not herein utilized since the first two positions adequately describe where the data is) and this data signal represents the bit identified as 00 on QA bus 614. The signal QA00110 is similar to the previous signal except the data identified by this signal is the data on position 01 of the QA bus 614. Thus by utilizing this convention and Figures 5 through 9 the ring protection hardware is fully defined and may be easily built by a person of ordinary skill in the computer art.

Referring to Figure 7b there is shown the detailed logic block diagram for UV register 503. Signal UVH0L10 is a hold signal for UV register 503 which is generated via inverter 703b when none of the one-legged AND gates 701b—708b has a high signal applied to it. UVH0L10 signal is applied to AND gate 723b and causes information stored in the UV register 503 to be held therein. Signal UVH0L1E coupled to the input of AND gate 704b and to the outputs of AND gates 705b—708b extends the number of control signals that may generate the hold signal UVH0L10. Signal UV0BK10 coupled to the outputs of AND gates 710b—713b and to the input of AND gate 722b is also utilized to extend the number of inputs signals that may be applied to flip-flop 724b. Signal UV1BK10 coupled to the outputs of AND gates 716b—718b and to the input of AND gate 727b similarly extends the number of input signals that may be applied to flip-flop 729b.

Referring now to Figure 7g there is shown the detailed logic block diagram of UO register 512. AND gates 701g—704g are OR'ed together and their output is applied as an input to inverter 705g. AND gates 706g—709g are also OR'ed together and their outputs are coupled to flip-flop 710g. Also one input of AND gate 709g is coupled to the U000010 terminal of flip-flop 710g. AND gates 711g—714g are also OR'ed together and are similarly coupled to flip-flop 715g. It will be noted also that an input of AND gate 706g is coupled to an input of AND gate 711g; an input of AND gate 707g is coupled to an input of AND gate 712g and an input of AND gate 709g is coupled

to an input of AND gate 714g. The UOH0L10 signal generated by inverter 705g is also coupled to an input of AND gate 709g and 714g and is utilized to hold information in the UO register 512. X00 represents a ground, whereas XNU means unused input.

Figure 7f is a detailed logic block diagram of UP register 501. It is similar to Figure 7g described supra except that different signals from different destinations and different sources are applied.

Referring now to Figure 7h there is shown the detailed logic block diagram of UM register 502. AND gate 701h—704h are OR'ed together to produce the UMH0L10 hold signal via inverter 705h. AND gates 706h—709h are OR'ed together and are coupled to the input of AND gate 704h in order to extend the range of signals that may be applied to produce the UMH0L10 hold signal. Similarly AND gates 711h—714h are OR'ed together and coupled to the input of AND gate 723h in order to extend the range of signals that may be applied to flip-flop 730h; and also AND gates 716h—719h are OR'ed together and are coupled to the input of AND gate 727h in order to extend the range of signals applied to flip-flop 731h. A line 740h for applying the PDA signals to flip-flop 730h and 731h is coupled at point 734h and 735h respectively. The input of AND gate 703h is also expanded to provide two further inputs URN1F00 and IRNUM10 by coupling the output of amplifier 733h to the input of AND gate 703h.

Referring now to Figures 7c—7e there is shown detailed logic block diagrams of write exception control logic 590, 1FU subcommand control logic 591, and read violation exception control logic 592 respectively. Referring first to Figure 7c there is shown flip-flops 705c and 710c which correspond to flip-flops 541 and 540 respectively. Under a micro-operation URW2F10 subcommand the information in flip-flop 710c is transferred to flip-flop 705c. The UWV1H10 hold signal is utilized to hold the information transferred to flip-flop 710c, whereas the UWV2H10 signal is utilized to hold the information transferred to flip-flop 705c. Similarly in Figure 7d information is transferred from flip-flop 710d to flip-flop 705d under micro-operation signal URNSW10, and in Figure 7e information from flip-flop 710e is transferred to flip-flop 709e under control of micro-operation signal URW2F10.

Referring now to Figures 8a, 8b and 8d there is shown detailed logic block diagrams of UWV logic 506, UWEP logic 507, and UMR logic 505 respectively. Referring first to Figure 8a there is shown logic for generating a high signal when one

of the test conditions 510 is true and also for generating the execute violation signal when the contents of UW register is less than or equal to the contents of UM register is less than or equal to the contents of UV register is not true. When the signal UWLEV10 is generated it indicates that the contents of UW register 504 is less than or equal to the contents of UV register 503. The logic for generating this signal was derived pursuant to the following Boolean expression:

$$X_1 = \overline{(BCD)} + (AB\bar{D}) \times (\bar{A}\bar{C})$$

Where X_1 represents the output of amplifier 805a and the various letters of the expression represent different input terminals of AND gates 801a—804a.

An indication that the contents of UV register 503 is greater than or equal to the contents of UM register 502 is had when UVGEM10 signal is generated. This signal is generated via inverter 820a in response to various inputs on AND gates 816a—819a which are OR'ed together and coupled to the input of inverter 820a. The logic for generating the UVGEM10 signal is made pursuant to the following Boolean expression:

$$X_2 = \overline{(BCD)} + (AB\bar{D}) + (\bar{A}\bar{C})$$

An indication that the contents of UM register 502 is greater than or equal to the contents of UV register 503 is indicated by generating signal UMGEV10 via inverter 810a in response to the various inputs of AND gates 806a—809a which are OR'ed together. The logic for generating this signal is derived from the following Boolean expression:

$$X_3 = \overline{(BCD)} + (AB\bar{D}) + (\bar{A}\bar{C})$$

(Wherein X_3 is the generated output signal).

Similarly the UVEQM10 signal is generated pursuant to the following Boolean expression:

$$X_4 = \overline{(A\bar{C})} + (\bar{A}\bar{C}) + (\bar{B}\bar{D}) + (\bar{B}D)$$

Generation of the UVEQUM10 signal indicates that the contents of the UV register 503 is equal to the contents of the UM register 502.

The generation of the UMGEW10 signal indicates that the contents of the UM register 502 is greater or equal to the contents of the UW register 504 and is generated pursuant to logic having the following Boolean expression:

$$X_5 = \overline{(BCD)} + (AB\bar{D}) + (\bar{A}\bar{C})$$

Generation of the UMGW10 signal indicates that the contents of UM register 502 is greater than the contents of UW register 504 and this signal is generated by logic defined by the following Boolean expression:

$$X_6 = (AB\bar{D}) + \bar{C}(B\bar{D} + A)$$

The generation of the UWGMV00 signal indicates that the contents of UW register less than or equal to the contents of UM register is not true. It is obtained when the UVGEM10 signal indicating that the contents of UV register is greater than or equal to the contents of the UM register, and the UMGEW10 signal indicating that the contents of the UM register is greater than or equal to the contents of the UW register are both high.

Referring now to Figure 8b a UMEQP10 signal is generated by logic derived from the following Boolean expression:

$$X_7 = \overline{(A\bar{C})} + (\bar{A}\bar{C}) + (\bar{B}\bar{D}) + (\bar{B}D)$$

When this signal is high it indicates that the contents of UM register 502 is greater than the contents of UP register 501.

Referring to Figure 8d there is shown the detailed logic block diagram for performing the operations of UMR logic 505 shown on Figure 5. One of the operations of this logic is to determine the maximum value of the contents of UP register 501 and of bits 2 and 3 of UBS logic 606. In order to do this there must be an indication whether contents of UP is less than the contents of UBS or the contents of UP is greater than the contents of UBS. The generation of UPBEB10 signal indicates that the contents of UP register 501 is less than or equal to bits 2 and 3 of UBS logic 606; whereas the generation signal UPGTB10 indicates that the contents of UP register 501 is greater than bits 2 and 3 of UBS logic 606. These signals are generated by logic which has been defined by the following Boolean expression:

$$X_8 = \overline{(BCD)} + (AB\bar{D}) + (\bar{A}\bar{C})$$

Where X_8 is the output of inverter 805d and the letters of the expression are various inputs of the AND gates 801d—803d.

To illustrate how the maximum value of the contents of UP register and UBS logic may be determined by the output signals UMPB010 and UMPB110 of amplifier 814d and 817d respectively, assume first that the contents of register UP are less than or equal to bits 2 and 3 of UBS logic because bit 2 is 1 and bit 3 is 1 whereas UB register

contains 01. This is indicated by the signal UPLEB10 being high and the signal UPGTB10 being low since it is the inverse of signals UPLEB10. This high UPLEB10 signal is applied to one input of AND gate 813d and also one input of AND gate 806d. If bit 2 of UBS logic is a 1 as indicated by signal UBS0210 then AND gate 813d is enabled and signal UMPB010 goes high and indicates that bit 2 on UBS logic is a 1. Moreover if bit 3 of UBS logic is a 1 indicated by input signal UBS0310 being applied as another input of AND gate 816d then AND gate 816d is enabled and signal UMPB110 is high or a 1. Therefore under the assumed conditions where bits (2, 3) UBS logic is greater or equal to the contents of UP register the maximum value of the two quantities is in UBS, and its number is binary 11 or decimal 4. Hence it is seen how a comparison is first made to determine which hardware contains the maximum, and then a determination is made as to the value of that maximum. By similar analysis one may see how the value of the UP register may be determined by signals UMPB010 and signals UMPB110 when the contents of UP register is greater than the second and third bit of UBS logic. Similarly the maximum value of UM register 502 or UV register 503 may be determined by signals UVGEM10 and UMGTV10 respectively, when UV register 503 is greater than or equal to UM register 502, and conversely when UM register 502 is greater than UV register 503.

Referring now to Figures 9a—9i a legend of symbols utilized in Figures 7 and 8 is shown. Figure 9a shows the symbol when there is a connection internally within the logic board. Figure 9b illustrates an output pin connection. Figure 9c indicates an input pin connection and is generally a source outside of the logic board illustrated. Figure 9d is the symbol utilized for an AND gate. Figure 9e is the symbol utilized for an amplifier; whereas Figure 9f is the symbol utilized for an inverter. Figure 9g illustrates three AND gates 901g—903g that are OR'ed together thus causing output 904g to go high when any one of AND gates 901g—903g is high. Figure 9h shows the symbol of a flip-flop having a 00 reset terminal and a 10 set terminal. A PDA line supplies the clock pulse for causing the flip-flop to switch states when other conditions are present on the flip-flop. Figure 9i represents a micro-operation control signal.

In order to enforce the ring protection scheme between procedures executing in different rings, the invention employs push-down stacks for its procedure linkage mechanism wherein a portion of each stack called a stack frame is dynamically

allocated to each procedure. Different stack segments are used for each ring with one stack segment corresponding to one ring. Thus when a procedure is executed in ring RN its stack frame is located in the RN stack segment. Referring to Figure 10 there is shown three stack segments 1001—1003, with each stack segment having stack frames S1—S3 respectively. Ring 3 is assigned to stack segment 1001, ring 1 assigned to stack segment 1002 and ring 0 is assigned to stack segment 1003. Within each stack segment there is a procedure 11 associated with stack frame S1 of segment 1001, a procedure P2 associated with stack frame S2 of stack segment 1002 and a procedure P3 associated with stack frame S3 of stack segment 1003. The segmented addresses (i.e. segment number and segment relative address SEG, SRA) of the first bytes of the stack segments for rings 0, 1 and 2 respectively are located in stack base words SBW0—SBW2 respectively which are in turn located in process control block 104. Since the ring 3 stack segment can never be entered by an inward call (i.e. from a ring higher than ring 3) its stack starting address is not needed. Each stack frame S1, S2, S3 is divided into a working area 1005, 1006, 1007 respectively; an unused portion 1008, 1009, 1010, which is utilized for alignment purposes; a register saving area 1011, 1012, and 1013; and a communication area 1014, 1015, and 1016 respectively. The working area is utilized by its procedure as needed and may contain material required by the process such as local variables, etc. The saving area of the stack frame is utilized to save the contents of various registers such as the status register, the T-register and the instruction counter contents ICC. The communications area stores information which is needed to pass parameters between procedures. Prior to a call to a given procedure the user saves those registers he wishes saved and moreover loads into the communication area the parameters to be passed to the called procedure. When the call is made, the hardware saves the contents of the instruction counter and other specified registers to facilitate a return from the called procedure. Each procedure call creates a stack frame within a stack segment and subsequent procedure calls create additional frames. Hence a stack is created and consists of a number of contiguous parts called stack frames which are dynamically allocated to each procedure. These stacks reside in stack segments. Generally the first stack frame is loaded into the beginning of the segment and succeeding frames are loaded after it. The last frame loaded is considered the top

5
10
15
20
25
30
35
40
45
50
55
60
65

70
75
80
85
90
95
100
105
110
115
120
125
130

of the stack. A T-register 114 on Figure 1, locates the top of the stack for the currently active process. A procedure such as for example P1 which is executing in ring 3 may call a procedure P2 executing in ring 1 which in turn calls a procedure P3 which is now executing in ring 0. As each procedure is called it creates within its ring stack segment a stack frame (i.e. defining the environment for the procedure execution) and the T-register 114 is loaded which gives the address of the top of the stack for the current active process. The procedure P1 (as previously assumed) may call procedure P2 which in turn may call procedure P3 and since these calls are from a higher ring number to a lower ring number a ring crossing entailing an inward call is required and is accomplished in a manner to be described infra. During each change of procedure the necessary registers and parameters are saved in order to facilitate a return from the called procedure.

A procedure is always accessed through a procedure descriptor 1110 by means of the ENTER PROCEDURE INSTRUCTIONS. The format of the ENTER PROCEDURE INSTRUCTION 1100 is shown on Figure 11a. The operation code (OP) 1101 occupies bit positions 0 through 7. The complementary code 1102 is a one bit code and occupies bit position 8 to 9; if the complementary code is set to logical 1 the instruction is ENT, whereas if the complementary code is logical 0 the instruction is ENTSR and the base register must be base register 0 (BRO). The address syllable AS 1104 occupies bit positions 12 thru 31 and provides the address syllable AS of the procedure descriptor 1110. When an ENTER PROCEDURE INSTRUCTION requires a ring crossing a gating procedure descriptor 1120 is obligatorily accessed. This is indicated by the GS field 1302 of segment descriptor 1301 being set to logical 10. Generally the GS field is set to 10 when one of the ENTER PROCEDURE INSTRUCTIONS is utilized. As described in the application No. 21630/76, Serial No. 1,465,344, the segment descriptor is utilized to point to the base of the segment desired, in this instance the segment 1300 containing gate procedure descriptors GPD 1120. The first word of the segment 1300 containing the gating procedure descriptors (GPD's) is formatted as shown in Figure 11c. The TAG 1121 occupies bit positions 0 and 1 and must indicate a fault descriptor i.e. the TAG field must be set to logical 11. The Caller's Maximum Ring Number CMRN 1122 occupies bit positions 2 and 3, and indicates the maximum ring from which a calling procedure through the gated procedure descriptor GPD is legal. A call

violation exception is generated if the caller's ring number is greater than CMRN 1122. The gated procedure descriptor address boundary GPDAB 1124 occupies bit positions 10 through 31 and it must be greater than the segment relative address SRA (i.e. the GPD's displacement in the segment of procedure descriptors 1300), otherwise an illegal GPD access exception occurs. Thus a gating procedure descriptor GPD is utilized as the first word of the segment containing procedure descriptors and is utilized to determine whether the caller has a right to access the segment via the caller's maximum ring number CMRN and whether or not the procedure descriptor called is within the gating procedure descriptor's address boundary. Once it is determined that there is a legal call to the segment and the caller has a right to enter the segment the address is obtained from the address syllable AS 1104 of enter instruction 1100 and the required procedure descriptor 1110 (see also Figure 13) is accessed. The format of procedure descriptor 1110 is shown on Figure 11b and is comprised of two 32 bit words—word 0 and 1 respectively. Word 0 contains the segmented address 1113 of the entry point EP of the procedure desired. The segmented address, as is the case with the segmented address of any operand, is comprised of the segment number SEG and the segment relative address SRA. Word 0 of the procedure descriptor includes an entry point ring number EPRN 1112 and a TAG field 1111. The value of the TAG is interpreted as follows:

- a. if the TAG contains logical 00 the procedure descriptor is direct;
- b. if the TAG is logical 01 the procedure descriptor is an extended descriptor and includes word 1 making a total of two words;
- c. if the TAG is logical 10 the procedure descriptor is indirect and an illegal procedure descriptor exception occurs; and
- d. if the TAG is logical 11 it is a fault procedure descriptor and an exception occurs.

Word 1 of the procedure descriptor is 32 bits long and is utilized when the TAG indicates an extended descriptor and contains the segmented address of a linkage section whose contents are loaded in base register BR 7 at procedure entry time.

Referring to Figure 12 a portion of the ENT instruction is shown and more specifically that portion which pertains to the ring crossing and ring checking requirements. The ENT instruction is called, 1201 and a comparison is made 1202 wherein the segmented part of the base register BRn is compared to the segmented part of the address of the T register, and if

5
10
15
20
25
30
35
40
45
50
55
60
65

70
75
80
85
90
95
100
105
110
115
120
125
130

they are not equal an illegal stack base register 1208 is indicated. If on the other hand they are equal another comparison 1203 is made wherein the 30th bit including the next two bits (i.e. bits 30 and 31) of base register, BR_n is compared to 0 and if it is not equal to 0, then once again an illegal stack base register 1208 is indicated. If it is equal to 0 it indicates that the contents of BR_n is aligned with respect to the word boundary and another comparison 1204 is performed to determine that the TAG of BR_n (i.e. the two bits starting from bit 0) is equal to 0. A TAG having a logical 0 indicates information is accessed via a direct descriptor which is one of the requirements of the ENT instruction. If the TAG (i.e. bits 0 and 1 of BR_n) is equal to 0 then the functions stated in flow charts of Figures 14 through 16 are performed (see flow chart Figure 12 block 1205). If these meet the necessary requirements a further check 1206 is made to determine whether the segment relative address of the entry point which was given (SRA_{EP}) is even, because instructions start on a half-word boundary. If it is not even then an illegal branch address exception is generated 1209 however if it is legal the ENT instruction is executed 1207 via further steps not shown.

Referring now to the flow charts of the access checking mechanism Figures 14—16, generally the following operations are performed each time the instruction ENTER PROCEDURE is issued:

- the caller's right to call the callee is checked by first determining from the second word of the segment descriptor the call bracket in which the caller is executing. (The call bracket is determined by taking the minimum ring number from the write ring number field WR and the maximum ring number from the maximum ring number field MAXR).
- a decision is made about the next process ring number by determining whether the caller is in the same call bracket as the callee, which implies don't do anything; whether the caller is in a call bracket requiring that he make an outward call in which case an exception condition is generated which is handled by a mechanism not described herein; or finally whether the caller is in a call bracket which requires an inward call (i.e. going to a call bracket which requires ring crossing from a larger ring number to a smaller ring number in which case the ring crossing must be at a valid entry point EP and the entry point must be validated).
- a stack frame is created for the callee (i.e. space in the aforementioned format of the appropriate segment is allocated), and the stack frame and the stack frame registers are updated;
- a branch to the entry point of the procedure pointed to by the procedure descriptor is performed.

Referring now to Figure 14 the access checking is started 1401 by obtaining the address syllable AS containing the effective address ring number EAR, the segment number of the procedure descriptor SEG_{PD}, and the segment relative address of the procedure descriptor SRA_{PD}. Having developed this information the procedure descriptor 1110 is fetched 1403 from (SEG_{PD}, SRA_{PD}) ignoring access rights to scratch pad memory. The procedure descriptor 1110 will yield the TAG which determines whether the descriptor is direct, extended, indirect, or a fault descriptor; the entry point ring number EPRN; the segment (SRA_{EP}) which contains the entry point and the segment relative address (SRA_{EP}) of the entry point. The TAG is tested 1404 to determine whether the descriptor 1110 is direct, extended, indirect or a fault descriptor by checking its field in accordance to the code hereinbefore described. Only a direct or extended procedure descriptor is legal. An indirect or fault descriptor is illegal and upon access invokes an exception mechanism not herein described. Once it is determined that a legal procedure descriptor has been accessed the actual call right checking begins at point A 1405.

Referring now to Figure 15 and continuing from point A 1405 the maximum ring number MAXR, the write ring number WR, and the execute permission bit EP of the segment containing the entry points SEG_{EP} are fetched; this information is contained in the segment descriptor for the segment containing the entry points (SEG_{EP}). The write ring number WR is compared to the maximum ring number MAXR 1503 and if the write ring number WR is greater than the maximum ring number MAXR the segment is nonexecutable and an execute violation exception 1513 occurs. If the write ring number WR is less than or equal to the maximum ring number MAXR then the execute permission bit EP is compared to logical 1 and if the EP bit is not logical 1 then once again an execute violation exception 1513 occurs; however if the EP bit is equal to one the effective address ring number EAR of the calling procedure is maximized with EPRN to give a new EAR₂—[MAX (EAR₁, EPRN)] where EAR₂ is the maximum of PRN as found in the instruction counter IC, and all ring numbers in base registers and data descriptors, if any, found in the path which leads to the procedure descriptor. The

effective address ring number EAR_2 is then compared 1506 to the maximum ring number $MAXR$ of the $MAXR$ segment descriptor of SEG_{EP} which is the maximum ring number at which a procedure may execute. If EAR_2 is greater than $MAXR$ the procedure call is an inward call which requires that the procedure be entered by a valid entry point and the access checking operation branch to point B 1507. The following checking operations are then performed:

- a. the SEG_{EP} is checked to determine if it is a legal gate segment; and,
- b. the caller's maximum ring number $CMRN$ is checked to determine if it is greater than or equal to the effective address ring number EAR of the caller.

If these conditions are not true then an illegal gate segment exception 1603 or call violation exception 1615 occurs.

Referring now to branch point B 1507 of Figure 16 the first check 1602 that is made is to determine whether or not the segment which contains the procedure descriptors is a gate segment. This is done by examining the Gating/Semaphore field GS of the segment descriptor pointing to the segment of procedure descriptors, to determine if it is set to logical 10. If the GS field of the segment descriptor of the segment containing procedure descriptors is set to 10 it is then a gate segment and the first word of the segment containing procedure descriptors is a gated procedure descriptor GPD 1120 of Figure 11C and Figure 13. The first word 1120 of the segment containing procedure descriptors is then fetched from address SEG_{PD} , 0 ignoring access rights to scratch pad memory. It will be noted that the TAG field of the first word 1120 of the segment containing procedure descriptor SEG_{PD} 1300 must be a logical 11 (Figure 13) which indicates it is a fault descriptor. Moreover the MBZ field must be set to zero. These conditions are checked by hardware/firmware (arithmetic logic unit) stop 1605 and if these conditions do not hold an illegal gate segment exception 1603 results. However if these conditions do hold a check 1606 is further made to determine that the segment relative address of the procedure descriptor SRA_{PD} 1110 is a multiple of 8. If the condition of step 1606 does not hold an illegal system object address exception 1613 results otherwise the next step 1607 is performed. Step 1607 checks to determine whether or not the segment relative address of the procedure descriptor SRA_{PD} is within the address boundary $GPDAB$ 1124 of the gated procedure descriptor 1120; if it is not within that address boundary it is an illegal procedure descriptor and an illegal GPD

gated procedure descriptor access exception 1614 occurs. However if it is within the address boundary of the gated procedure descriptor (i.e. SRA_{PD} is less than $GPDAB$) then the caller's right to call the callee is checked 1608. This is performed by comparing the effective address ring number EAR_2 to the caller's maximum ring number $CMRN$ 1122 as found in the first word 1120 of the segment of procedure descriptors 1300. If EAR_2 is greater than the caller's $CMRN$ a call violation exception 1615 occurs which indicates that the caller in this particular instance has no right to legally call inward i.e. from a higher ring number to a lower ring number. On the other hand if EAR_2 is equal or less than $CMRN$, then the inward call is legal and a check is made 1609 to determine that the process ring number PRN which is the current process ring number found in the instruction counter IC just before the call was made is less than the maximum ring number $MAXR$ of SEG_{EP} ; and if it is the accessing mechanism branches to point C 1508, otherwise a new process ring number $NPRN$ is calculated and set to a maximum ring number $MAXR$ 1611. Generally the effective address ring number EAR_2 is the same as the process ring number PRN of the caller. Sometimes however, in cases where it is necessary to give maximum assurance that the caller will not be denied access to a given segment the EAR_2 is greater than the PRN . In those cases PRN is forced to take the value of EAR_2 in order to make sure that the call is returned to the maximum ring number upon an exit. To this point it will be noted that this checking mechanism was invoked because the EAR_2 was greater than the $MAXR$ hence greater than the top of the call bracket of the procedure and hence an inward call was necessary which necessitated going through a valid gate, and the mechanism included these gating checks. By branching back to C 1508 (Figure 15) a further check 1509 is made to determine then that the process ring number PRN is greater than the write ring number WR of SEG_{EP} which in this context is the minimum ring number at which a procedure may execute. If the write ring number WR is greater than the process ring number PRN an outward call exception 1514 occurs. However if WR is less than or equal to PRN the call is legal and $NPRN$ is set to PRN 1510.

Having made the above checks the inward call is made, and after performance of the desired operation a return back to the original point of the program in execution is made by the $EXIT$ INSTRUCTION. During the $ENTER$ INSTRUCTION the instruction counter IC

5

10

15

20

25

30

35

40

45

50

55

60

65

70

75

80

85

90

95

100

105

110

115

120

125

130

was saved in the saving area of the caller's stack frame before making the call. Moreover the caller's ring number was also saved during the ENTER INSTRUCTION and this was saved in base register 0 BRO.

The format of the EXIT INSTRUCTION 1130 is shown on Figure 11D. The operation code OP 1131 is found in bit positions 0—7 and the complementary code C 1133 is found in bit positions 12—15. The complementary code allows other instructions to use the same 8 bit op code. The MBZ field 1132 in bit positions 8—11 must be 0 otherwise an illegal format field exception occurs. (BRO is generally a pointer to the communications area of the caller's stack frame).

In performing the EXIT INSTRUCTION it is necessary to perform predetermined checks in order to ascertain that the caller didn't change his image which would permit him to operate a different privilege than was intended. Referring to Figure 17 the first check performed 1701 is to determine if the TAG of the instruction counter content (ICC) indicates a direct descriptor. A logical 00 in the TAG field indicates that it is direct if it is not an illegal stack data exception 1702 occurs, whereas if it is equal to 0 the ring field in the instruction counter content ICC is set to the new process ring number NPRN 1703. This sets the new process ring number NPRN to what it used to be when the call was first made. However further checks are made in order to ascertain that there was no further cheating. Hence the base register 0 ring number located at bit position 2 and extending for 2 bit positions from and including bit position 2 must be equal to the new process ring number NPRN 1704. (It will be recalled that when the ENTER INSTRUCTION was called the ring number of the caller before the call was made was stored in bits 2 and 3 of base register 0 (BRO). If check 1704 indicates that the new process ring number NPRN is not

equal to the ring number in bit positions 2 and 3 of the base register 0 (BRO) an illegal stack data exception 1702 occurs. The next check 1705 determines whether an inward or an outward return must be performed. Since an inward call was previously performed an outward return is implied in order to reach the original point from which the procedure was called. Moreover since the invention does not permit an outward call there is never a necessity to return inward. Hence the new process ring number NPRN is compared to the process ring number PRN 1705, and if NPRN is less than PRN an inward return is implied and an inward return exception 1706 is generated. However if check 1705 is passed successfully (i.e. NPRN is greater or equal to PRN) then a check is made to determine that a return is made to the segmented address SEGr that called the procedure and a return to the call bracket of the calling procedure is made and moreover that the execute bit EP is set. This is performed by fetching the segment descriptor SEGr of the calling procedure 1707 and making checks 1709, 1711, 1712. In performing checks 1709, 1711, 1712, check 1709 and 1711 determine that the new process ring number NPRN is greater than the minimum ring number WR but less than the maximum ring number MAXR (i.e. that the ring number is in the call bracket of the calling procedure where it should be). Finally check 1712 makes sure that the execute permission bit EP is set to 1. Thus a full cycle is concluded a call was performed via an ENTER INSTRUCTION; the required operation or processing was performed via the called procedure; then a return via an EXIT INSTRUCTION to the calling procedure was performed.

Having shown and described the preferred embodiment of the invention, those skilled in the art will realize that many variations of modifications can be made to produce the described invention and still be within the scope of the claimed invention.

Glossary of Terms

- JOB—The job is the major unit of work for the batch user. It is the vehicle for describing, scheduling, and accounting for work he wants done.
- JOB STEP—A smaller unit of batch work. It is generally one step in the execution of a job consisting of processing that logically belongs together.
- TASK—The smallest unit of user-defined work. No user-visible concurrency of operation is permitted within a task.
- PROGRAM—A set of algorithms written by a programmer to furnish the procedural information necessary to do a job or a part of a job.
- PROCESS GROUP PLEX—The system's internal representation of a specific execution of a job.
- PROCESS GROUP—A related set of processes, usually those necessary for performance of a single job step.
- PROCESS—The controlled execution of instructions without concurrency. Its physical representation and control are determined by internal system design or convention.

Glossary of Terms (cont.)

- PROCEDURE—A named software function or algorithm which is executable by a computational processor without concurrency. Its physical representation (code plus associated information, invocation, and use are determined by internal system or designed convention).
- 5 LOGICAL PROCESS—The collection of hardware resources and control information necessary for the execution of a process.
- ADDRESS SPACE (SEGMENTATION)—The set of logical addresses that the CPU is permitted to transform into absolute addresses during a particular process. Although a processor has the technical ability of addressing every single cell of timing memory, it is desirable to restrict access only to those cells that are used during the process associated with the processor.
- 10 LOGICAL ADDRESS—An element of the process address space such as for example segment number SEG and Displacement D.
- BASIC ADDRESS DEVELOPMENT—A hardware procedure which operates on a number of address elements to compute an absolute address which is used to refer to a byte location in core.
- 20 PROCESS CONTROL BLOCK—A process control block PCB, is associated with each process and contains pertinent information about its associated process, including the absolute address of tables defining the segment tables the process may access.
- J. P. TABLES—A collection of logical addresses for locating a process control block associated with a process.
- 25 SEG_{PD}—The segment which contains the procedure descriptor.
- SEG_{EP}—The segment which contains the entry point, as found in the procedure descriptor.
- PRN—The process ring number, found in the instruction counter IC just before the call, or calculated by the ENTSR instruction.
- 30 EAR—The effective address ring number which is the maximum of:
(a) the process ring number PRN as found in the IC; or
(b) all ring numbers in the base register and data descriptors (if any) found in the path which leads to the procedure descriptor from the call instruction, including the entry point ring number EPRN located in the procedure descriptor itself.
- 35 MAXR—The maximum ring number at which a procedure may execute; MAXR is found in the segment descriptor of SEG_{EP}.
- WR—The minimum ring number at which a procedure may execute; WR is found in the segment descriptor of SEG_{EP}.
- 40 EP—Execution permit bit found in the segment descriptor of SEG_{EP}.
- CMRN—The caller's maximum ring number, as found in the first word of the segment SEG_{PD} if this segment is identified as a gate segment (i.e. with the code "gate" set).
- 45 NPRN—New process ring number.
- EPRN—Entry point ring number (found in the process procedure descriptor).

Addendum

Signal Name	Type	Function
(1) WSCLR	Control	Clears register to which it is connected.
(2) PDARG	Control	Clock Signal PDA.
50 (3) PDURGIT	Connecting	Pin connected to PDA at one end and resistor at the other.
(4) UWOBK	Connecting	Expands inputs to UW register.
(5) UWHOL	Control	Holds information in register to which it is connected.
55 (6) UW1BK	Control	Same as UWOBK but is connected to different input terminal of UW register.
(7) UW0000		Reset terminal of one flip-flop of register UW.
(8) UW0010		Set terminal of flip-flop of register UW.
60 (9) UW00100 UW00110		Same as 7+8 but different flip-flop.
(10) UVSPS	Control	Spare Control Input.

Addendum (cont.)

	Signal Name	Type	Function
	(11) UVSPD	Data	Spare Data Input.
5	(12) UVOBK	Expander	Same as UWOBK and UWIBK, but it connects different registers and gates.
	(13) UV00000 UV00010 UV00100 UV00110		Same as UW00000, UW00010, UW00100, UW00110, but applies to flip-flop UV.
10	(14) UWV1S	Control	Control input for UWV1F.
	(15) UWV1D	Data	Data input for UWV1F.
	(16) UWV2F	F/F	Write control flip-flop.
	(17) UWV1S UWV2S	Control	Control unit for UWV1F, UWV2F.
15	(18) UWV1D	Data	Data input for UWV1F.
	(19) UWV1H	Control	Hold UWV1F flip-flop.
	(20) UWV1C	Control	Clear UWV1F.
	(21) UWV2C	Control	Clear UWV2F.
20	(22) URN1S URN2S	Control	Control inputs for URN1F, URN2F.
	(23) URN1D	Data	Data Input for URN1F.
	(24) URNSW	Control	Transfer URN1F to URN2F and URN2F to URN1F.
25	(25) URN2F	F/F	Control loading max (UP, UBS2, 3 to UM).
	(26) URN1H	Control	Hold URN1F flip-flop.
	(27) URN2C	Control	Clear URN2F.
	(28) URW1S URW2S	Control	Control inputs for URV1F, URV2F.
30	(29) URW1D	Data	Data Input for URV1F.
	(30) URV2F	F/F	Read control flop.
	(31) XNU		Indicates terminal not used herein.
	(32) XOO		Grounded Input.

WHAT WE CLAIM IS:—

1. An internally programmed data processing apparatus CPU having a virtual memory system, and being responsive to internally stored instruction words for processing information and having stored in said virtual memory system a plurality of different types of groups of information each information group-type associated with an address space bounded by a segment having adjustable bounds, and comprising means for protecting the information in said-virtual memory system from unauthorized users by restricting accessibility to the information in accordance to levels of privilege, said means comprising in combination with an access checking mechanism;

(a) first means arranged in operation to store in said virtual memory system at least one segment table comprising a plurality of segment descriptors with each segment descriptor being associated with a predetermined one of said segments and each segment descriptor having a predetermined format containing an access information element and a base address element in predetermined positions of said format, said base address element being used for locating in said virtual memory system the starting location of a selected

one of said segments, and said access information element for specifying the minimum level of privilege required for a predetermined type of access that is permitted in a selected one of said segments;

(b) a plurality of second means having a predetermined format, communicating with said first means, arranged to store in a predetermined portion of said second means, a segment number SEG for identifying a segment table and the location of a segment descriptor within said segment table, said second means also being arranged to store in a predetermined other portion of said second means, an offset address within the segment identified by said segment descriptor said offset address locating from said segment base the first byte of a word within said segment;

(c) third means responsive to an address syllable element of an instruction being executed for addressing one of said plurality of second means;

(d) fourth means arranged to store a displacement from said address syllable,

(e) fifth means, communicating with said first, second, third and fourth means, arranged to add the displacement D and said base address to said offset; and,

(f) sixth means responsive to said access

information element in a selected one of said segment descriptors, restricting the accessibility to the segment associated with said selected one of said segment descriptors in accordance to the level of privilege and the type of access specified in said access information element, wherein each group-type of information is associated with a predetermined ring number indicative of a level of privilege said level of privilege decreasing as the associated ring number increases comprising means for determining the maximum effective address ring number EAR (i.e. minimum level of privilege) of a selected process to access a selected group of information, said means comprising;

(a) first means to store first information indicating the maximum ring number RD (i.e. minimum level of privilege) required to read information from said selected group;

(b) second means to store second information indicating the maximum ring number WR (i.e. minimum level of privilege) required to write information into said selected group;

(c) third means to store third information indicating the maximum ring number MAXR (i.e. minimum level of privilege) required to process information from said selected group; and,

(d) fourth means communicating with said first, second and third means, to determine the maximum of the contents of said first, second and third means whereby the effective address ring number EAR is generated.

2. Apparatus according to claim 1, wherein said second means for storing the maximum ring number WR additionally indicates the minimum ring number WR (i.e. maximum level of privilege) required to process information from said selected group.

3. Apparatus according to claim 1 or claim 2, wherein said fourth means to generate the effective address ring number comprises a comparator for comparing binary numbers.

4. Apparatus according to any one of claims 1 to 3 wherein the sixth means restricting the accessibility to the segment includes comparator means, communicating with said second means, to compare the effective address ring number EAR with the write ring number WR, and further including means communicating with said comparator means to generate a write-violation-exception signal when EAR is greater than WR.

5. Apparatus according to claim 4, wherein the sixth means restricting the accessibility to the segment includes seventh means, communicating with said second and third means thereof to compare the maximum ring number MAXR and the write ring number WR with the effective address ring number EAR, and further including eighth means, communicating with said seventh means for generating an execute-violation-exception signal when the MAXR is not equal or greater than EAR which in turn is not equal or greater than WR.

6. Apparatus according to claim 5, wherein in that the sixth means restricting the accessibility to the segment includes ninth means, communicating with said first means, for comparing the effective address ring number EAR with the read ring number RD, and further including tenth means, communicating with said ninth means, to generate a read-violation-exception signal when EAR is greater than RD.

7. Apparatus according to claim 6, wherein in that the sixth means restricting the accessibility to the segment includes eleventh means to store a process ring number PRN of a currently executing process, and also including twelfth means to communicate with said eleventh means, and further including thirteenth means communicating said said twelfth means for overriding said read-violation-exception signal when the effective address ring number EAR is equal to the process ring number PRN of the currently executing process.

8. Apparatus according to any one of the preceding claims wherein the access checking mechanism supervises transfer of control of said CPU from a first selected procedure (i.e. caller) having a first ring number indicative of a minimum level of privilege associated with said caller, to a second selected procedure (i.e. the callee) having a second ring number associated with said callee indicative of a minimum level of privilege associated with said callee, said access checking mechanism comprising

(a) first means for checking the caller's right to call the callee;

(b) second means, communicating with said first means, to compare the caller's ring number to the callee's ring number;

(c) third means responsive to said second means to permit a transfer of control of said CPU from said caller to said callee when the ring number of the caller is greater than the ring number of callee (i.e. inward call); and,

(d) fourth means also responsive to said second means to deny a transfer of control of said CPU from said caller to said callee when the ring number of said caller is less than the ring number of the callee (i.e. outward call).

9. Apparatus according to claim 8, wherein the access checking mechanism includes a plurality of ring stack-segment means each of said ring stack-segment means having associated with it a ring stack-segment number, indicative of the minimum level of privilege required by a selected one of said procedures to access a selected one of said ring stack segments.

10. Apparatus according to claim 9 wherein there are four ring stack segment means having ring numbers 0 to 3 respectively.

11. Apparatus according to claim 9 or claim 10 wherein the access checking mechanism includes stack-frame-element means associated with selected ones of said procedures, said stack-frame-element means being grouped within said ring stack-segment means in accordance with the ring number of the associated procedure of said

stack-frame-element means, said stack frame element means to save said register of said caller prior to passing control to said callee.

12. Apparatus according to claim 11, wherein the access checking mechanism includes first sub-element means, responsive to said first, second, third and fourth means, for communicating between a selected one of said stack-frame-means in a first ring stack-segment being associated with one ring number, and a selected other of said stack-frame-means in a second ring stack-segment associated with another ring number.

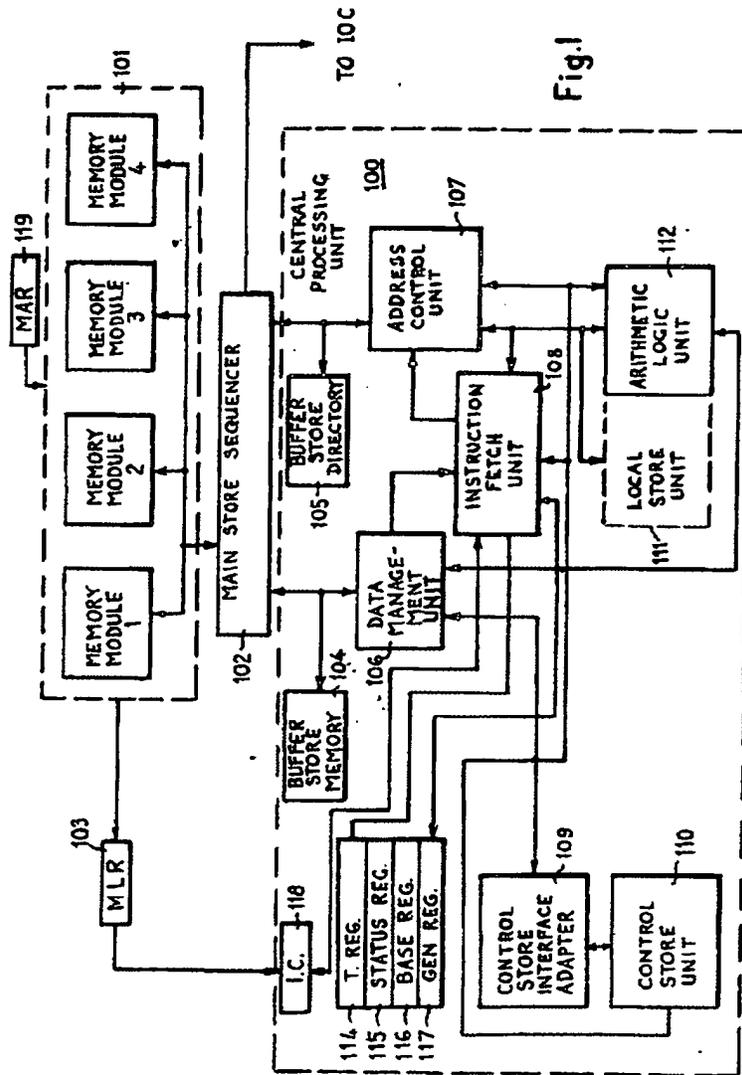
25

30

35

BARON & WARREN,
16, Kensington Square,
London, W8 5HL.
Chartered Patent Agents.

Printed for Her Majesty's Stationery Office, by the Courier Press, Leamington Spa, 1977
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.



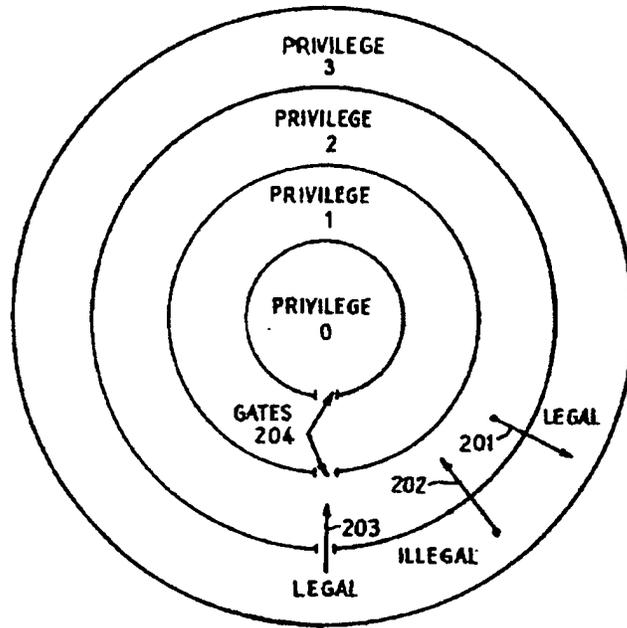
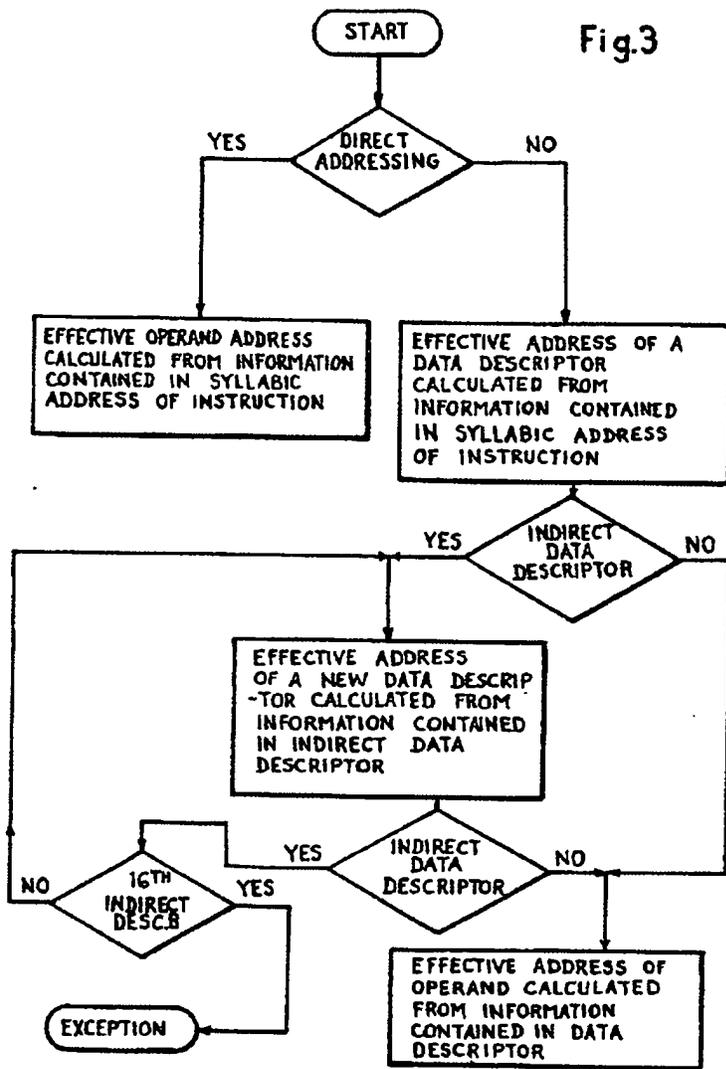


Fig.2

Fig.3



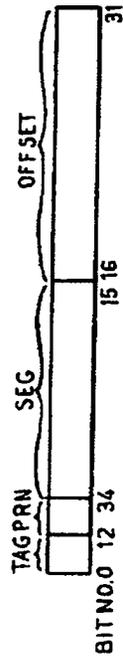


Fig. 4 A

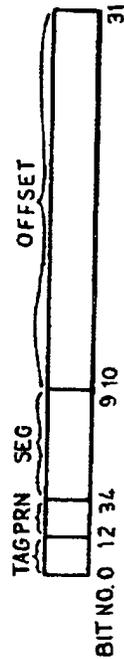


Fig. 4 B

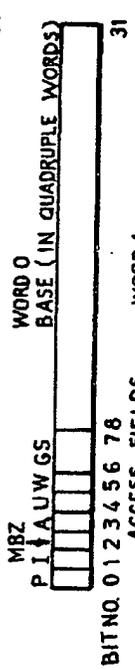


Fig. 4 C

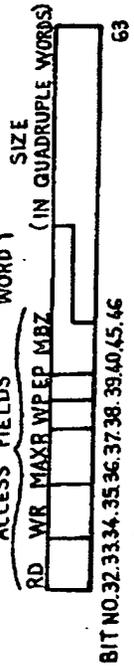


Fig. 4 D

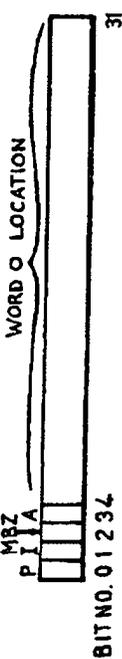


Fig. 4 E

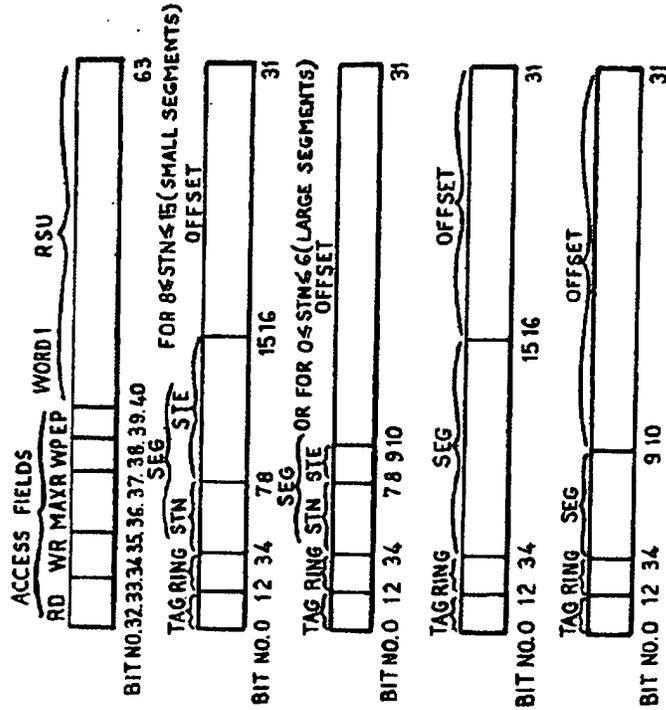


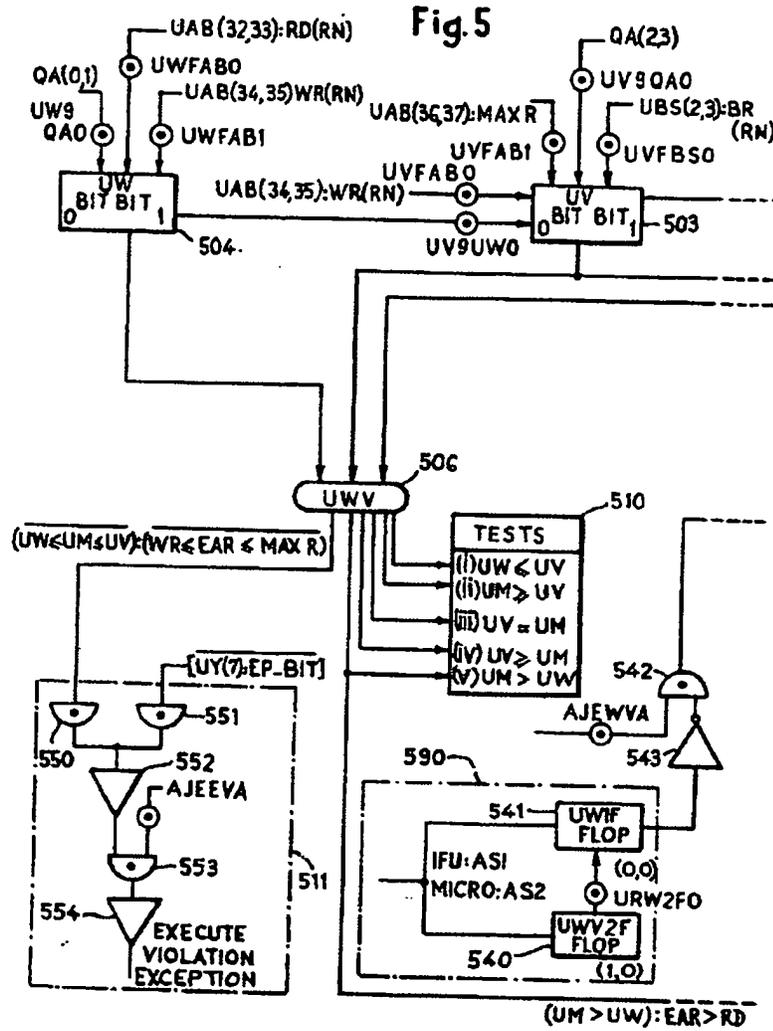
Fig. 4F

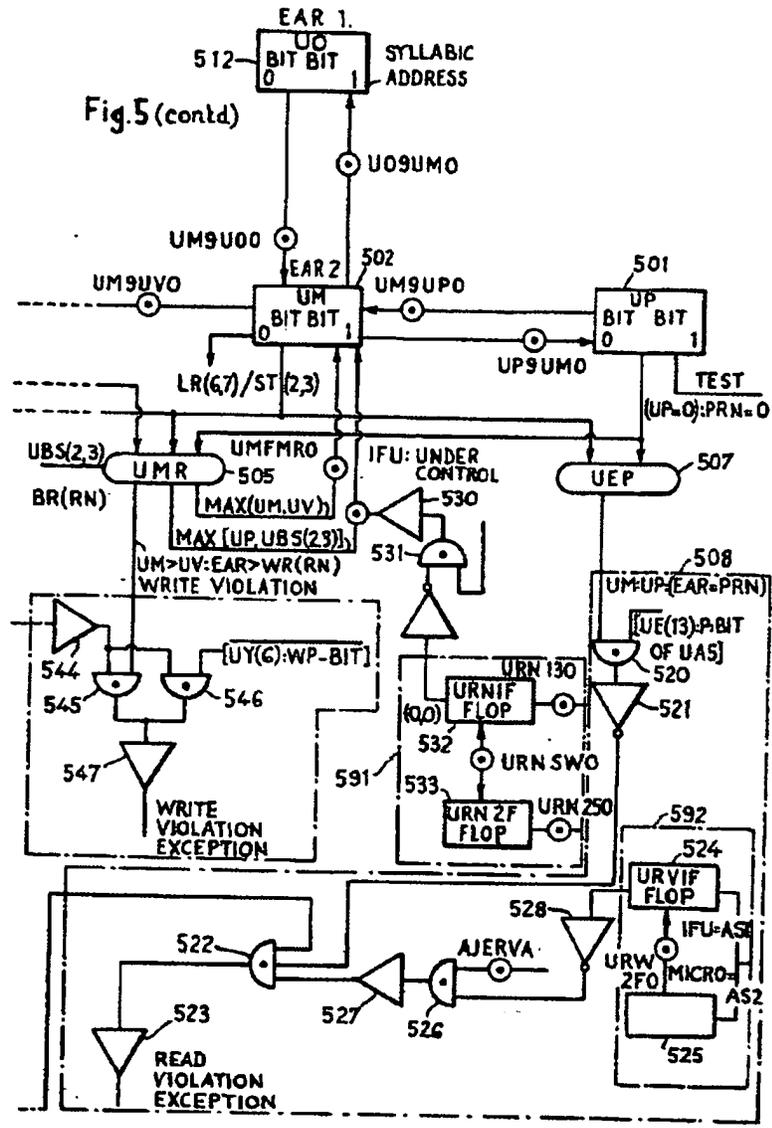
Fig. 4G

Fig. 4H

Fig. 4I

Fig. 4J





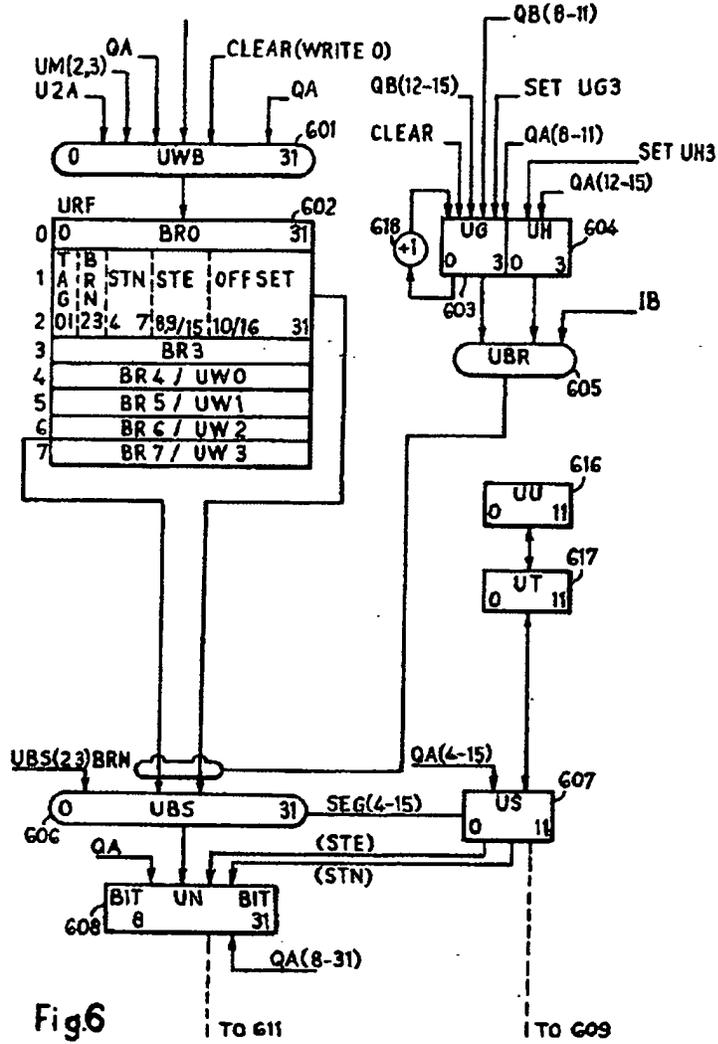


Fig 6

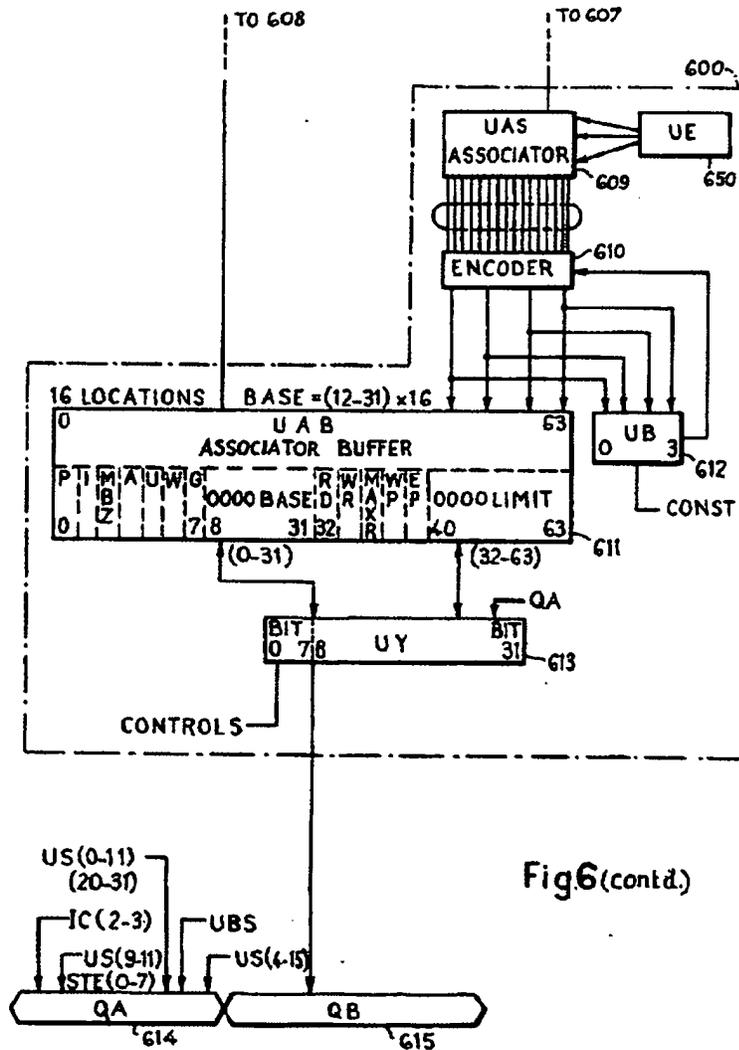
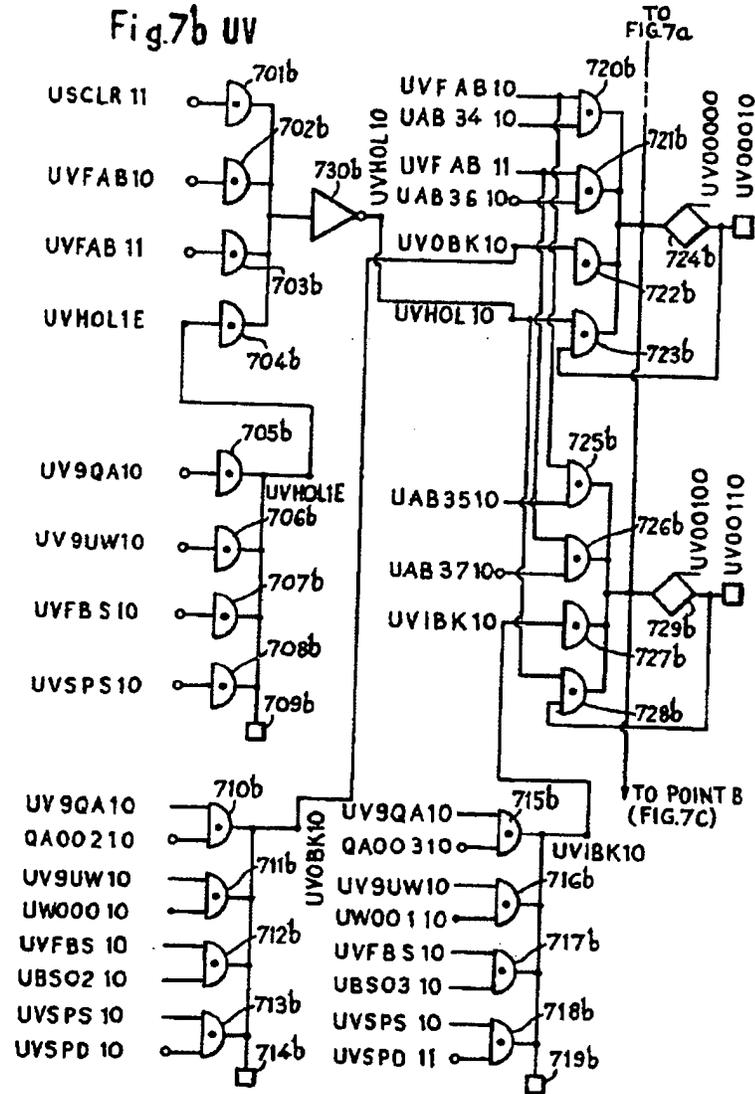


Fig 6(contd.)



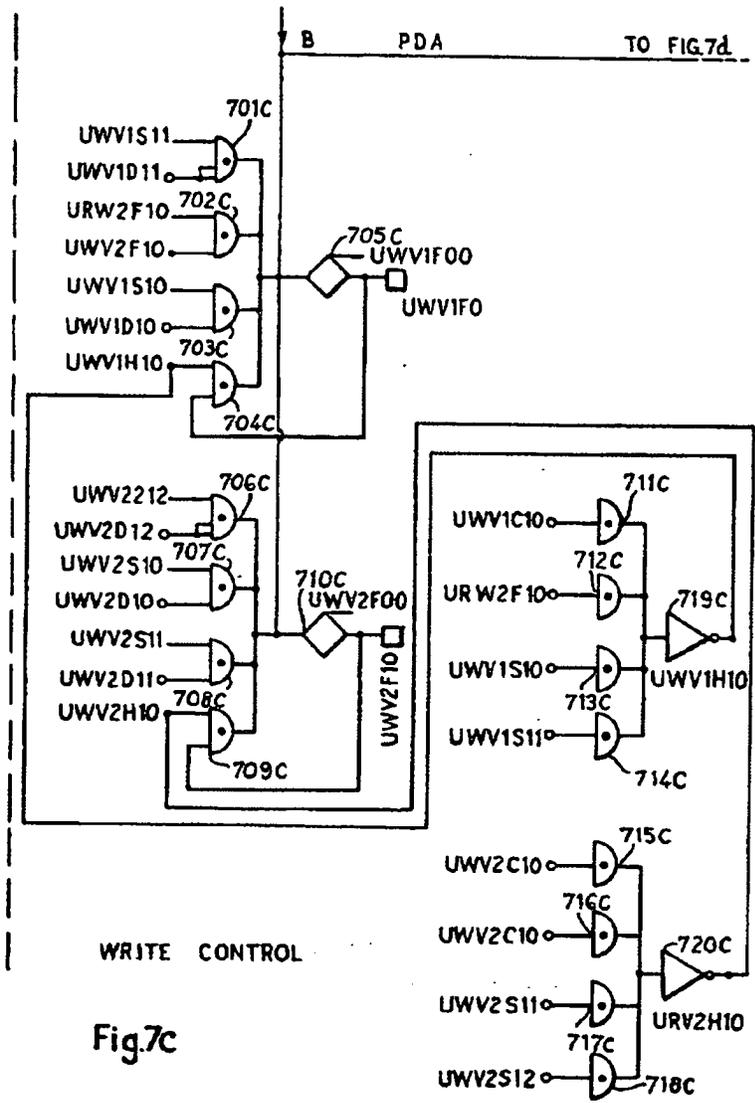


Fig. 7c

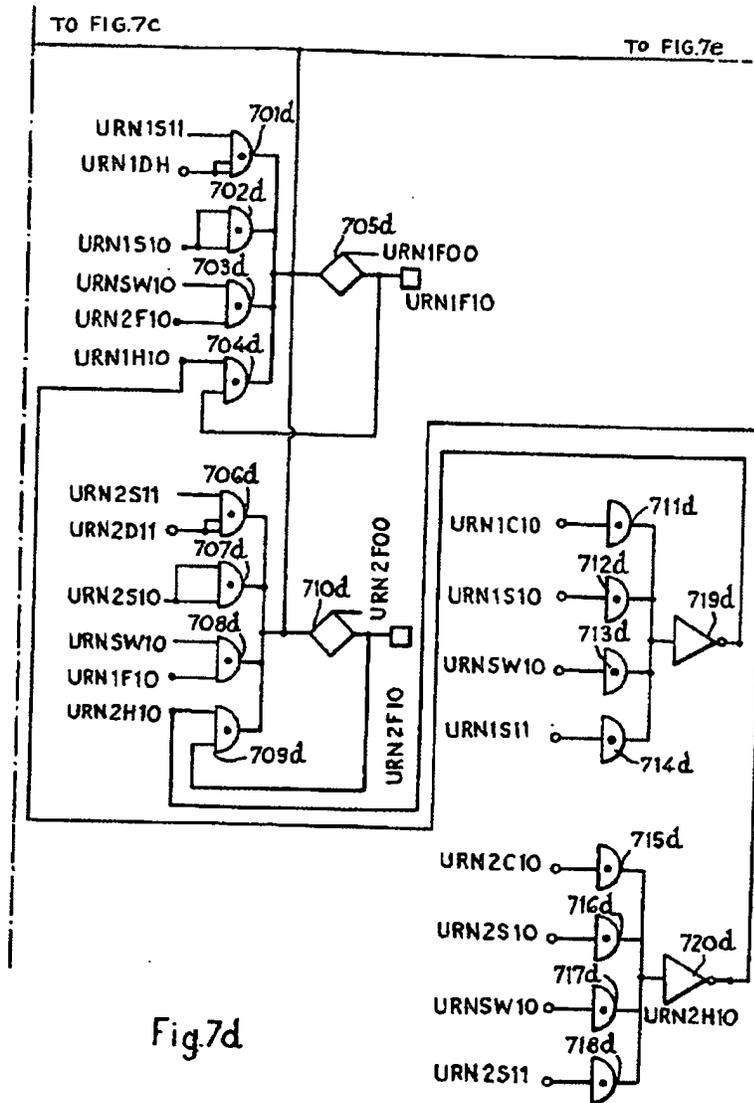
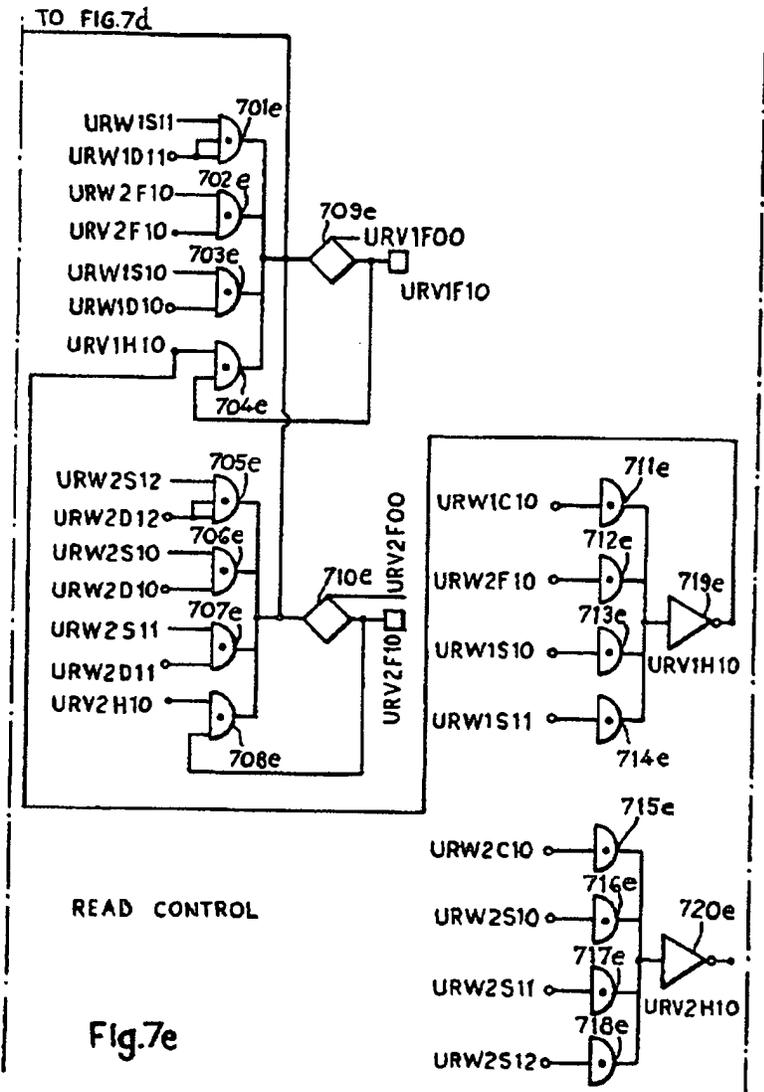


Fig.7d



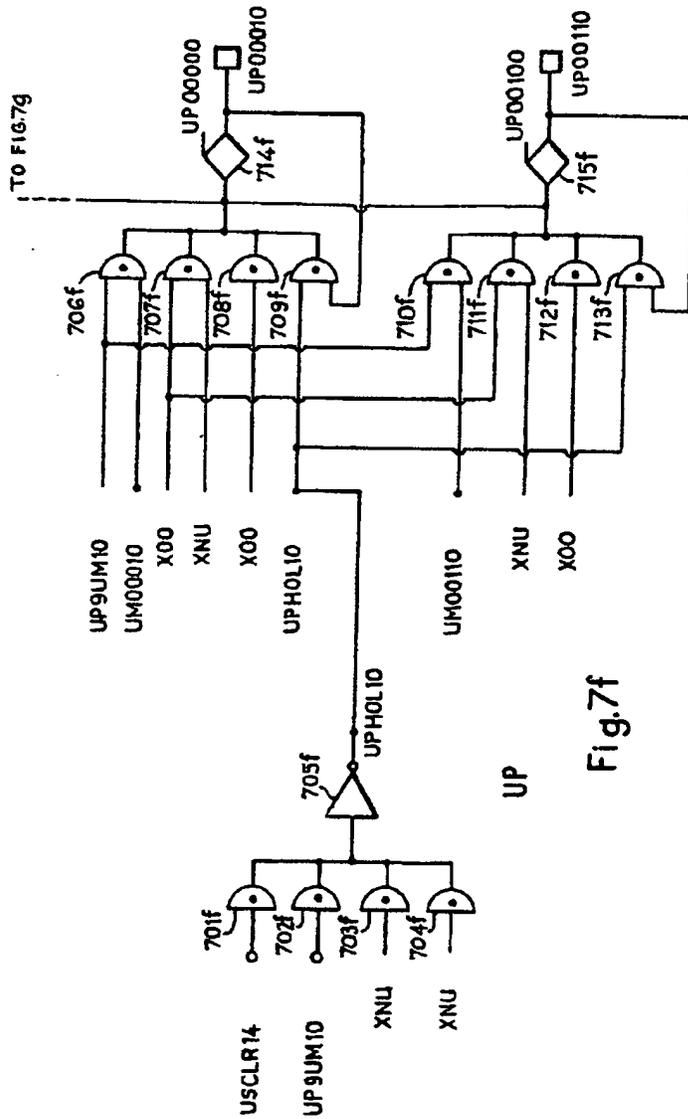


Fig.7f

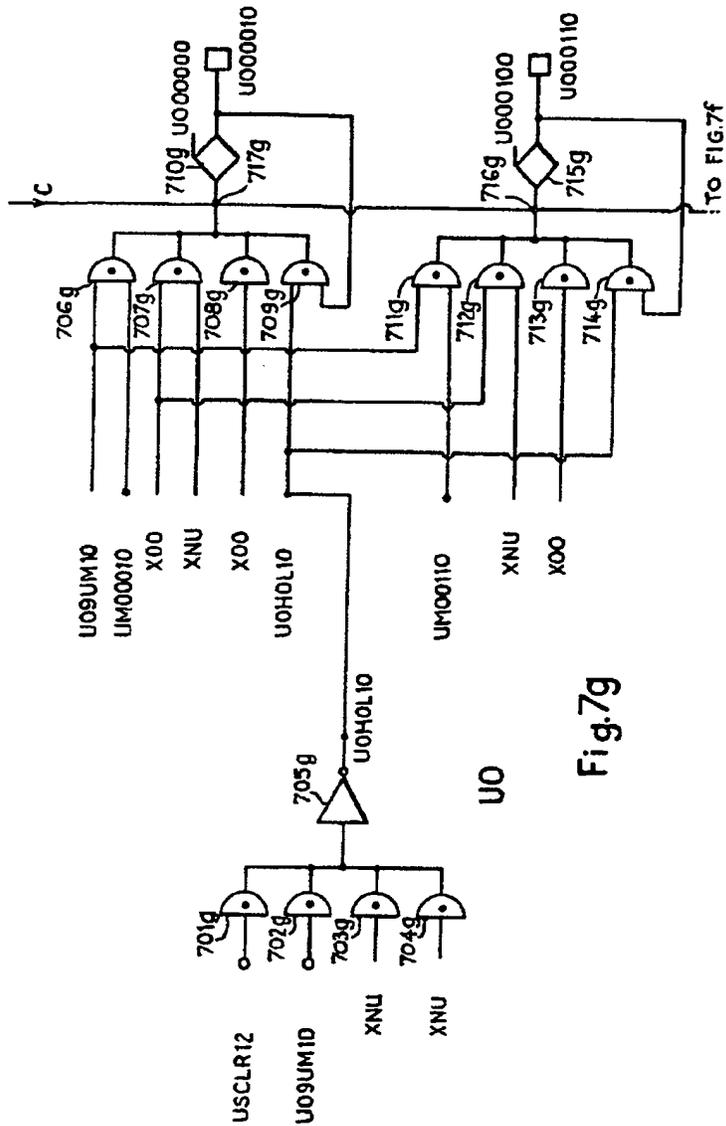
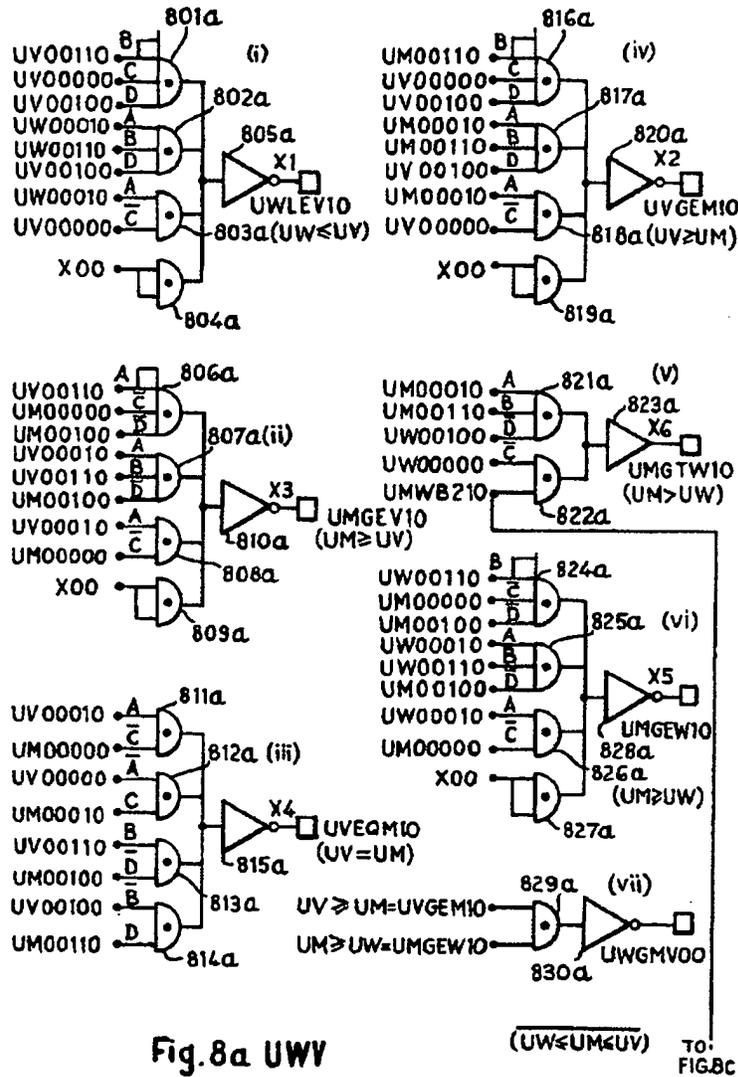


Fig. 7g



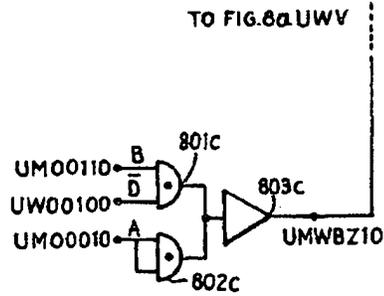


Fig.8c

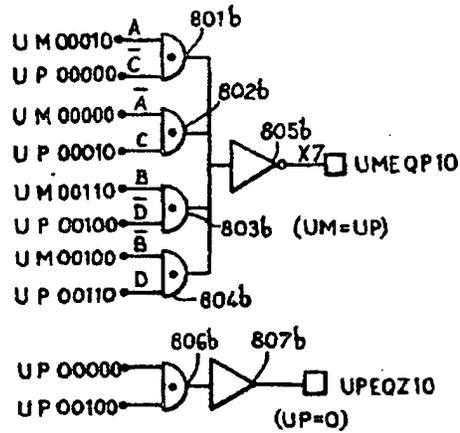


Fig.8b UEP

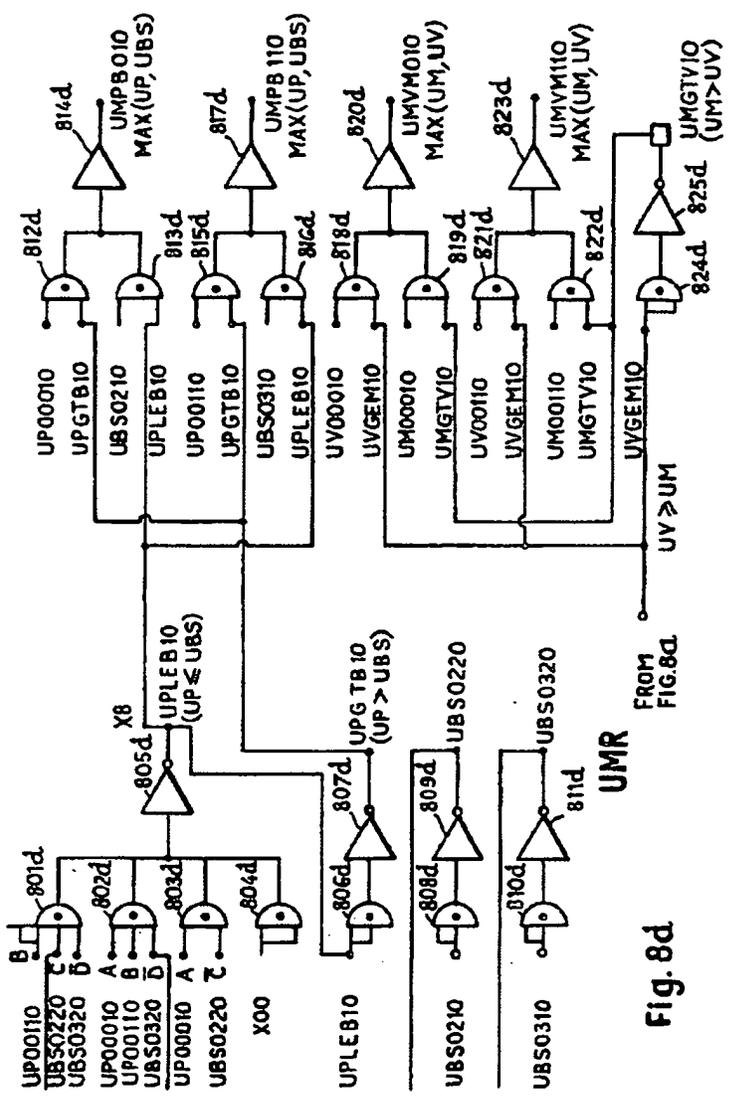
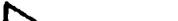
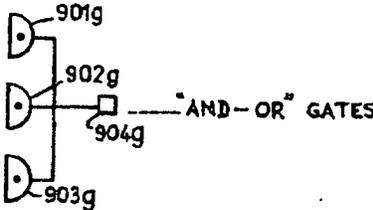
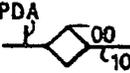


Fig. 8d

KEY TO SYMBOLS

- Fig. 9a  INTERNAL SIGNAL SOURCE
- Fig. 9b  OUTPUT PIN
- Fig. 9c  INPUT PIN
- Fig. 9d  AND GATE
- Fig. 9e  AMPLIFIER
- Fig. 9f  INVERTER
- Fig. 9g  AND-OR GATES
- Fig. 9h  FLIP-FLOP
- Fig. 9i  MICRO-OPERATION
- Fig. 9j X::Y-----X::Y
- Fig. 9k a:: θ -----START OF BIT α WHERE THERE
 ARE β BIT POSITIONS INCLUDING
 BIT α

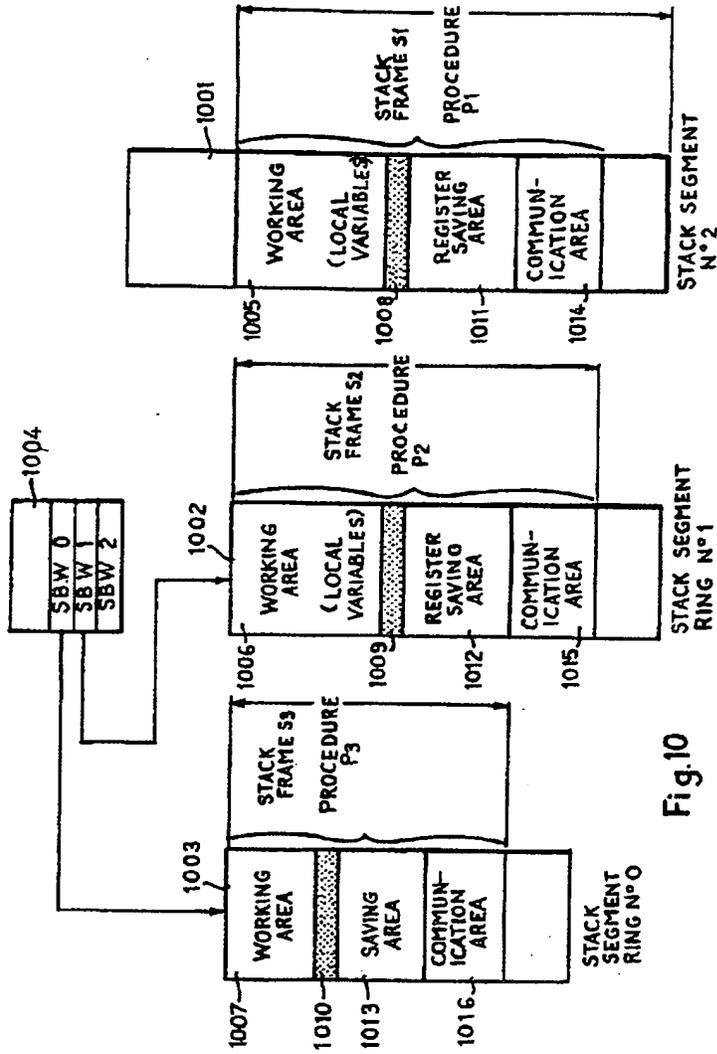


Fig. 10

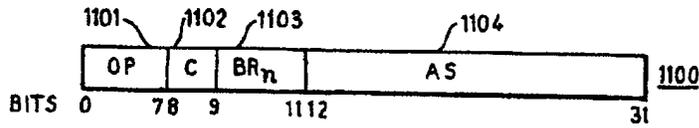


Fig. 11A

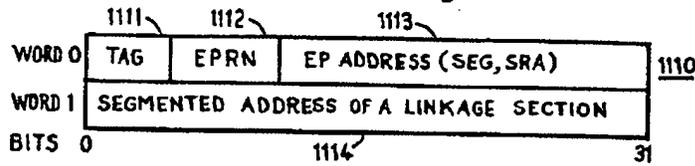


Fig. 11B

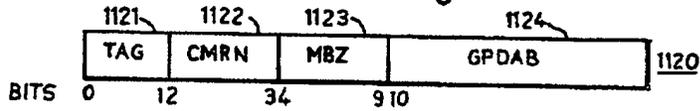


Fig. 11C

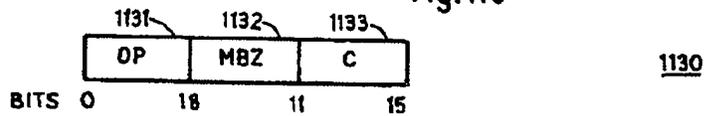


Fig. 11D

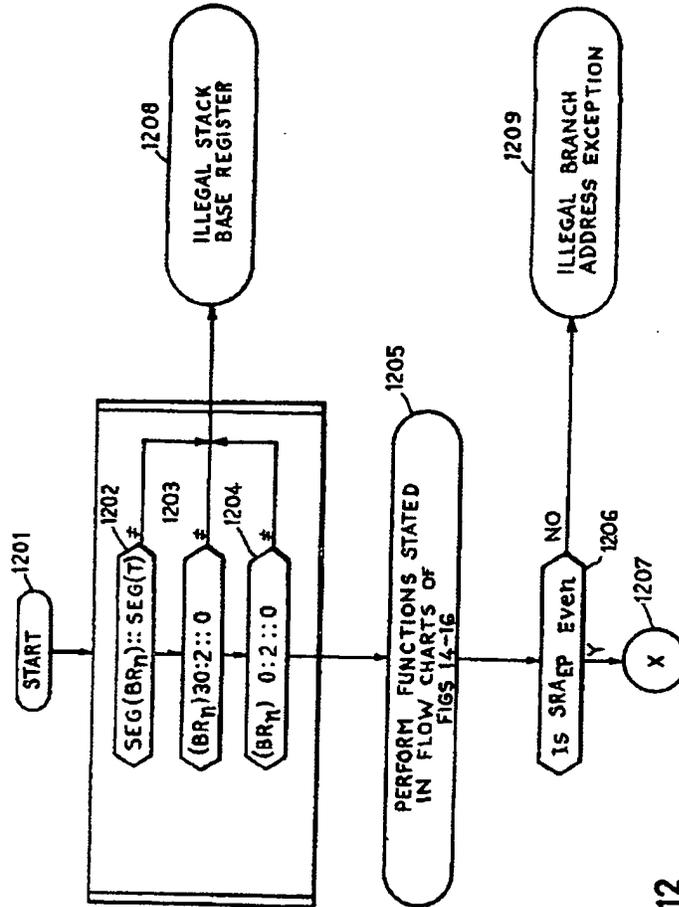


Fig. 12

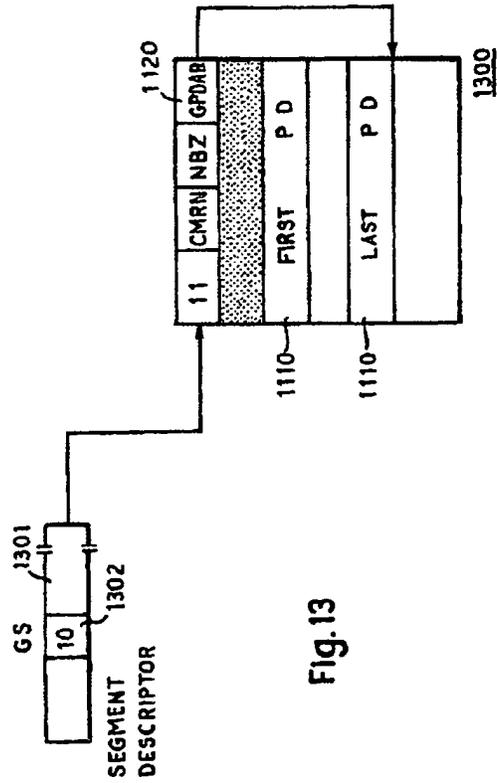


Fig. 13

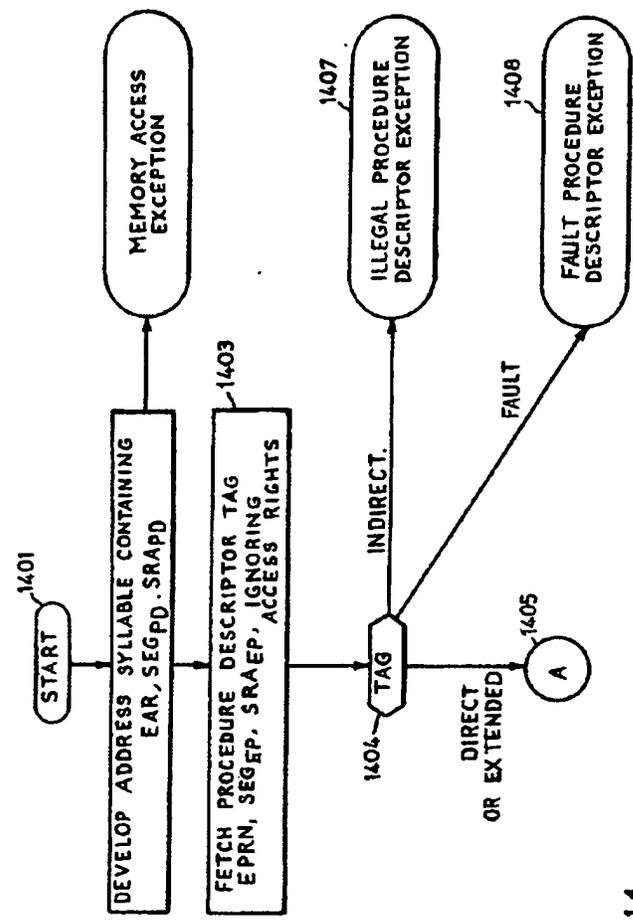


Fig.14

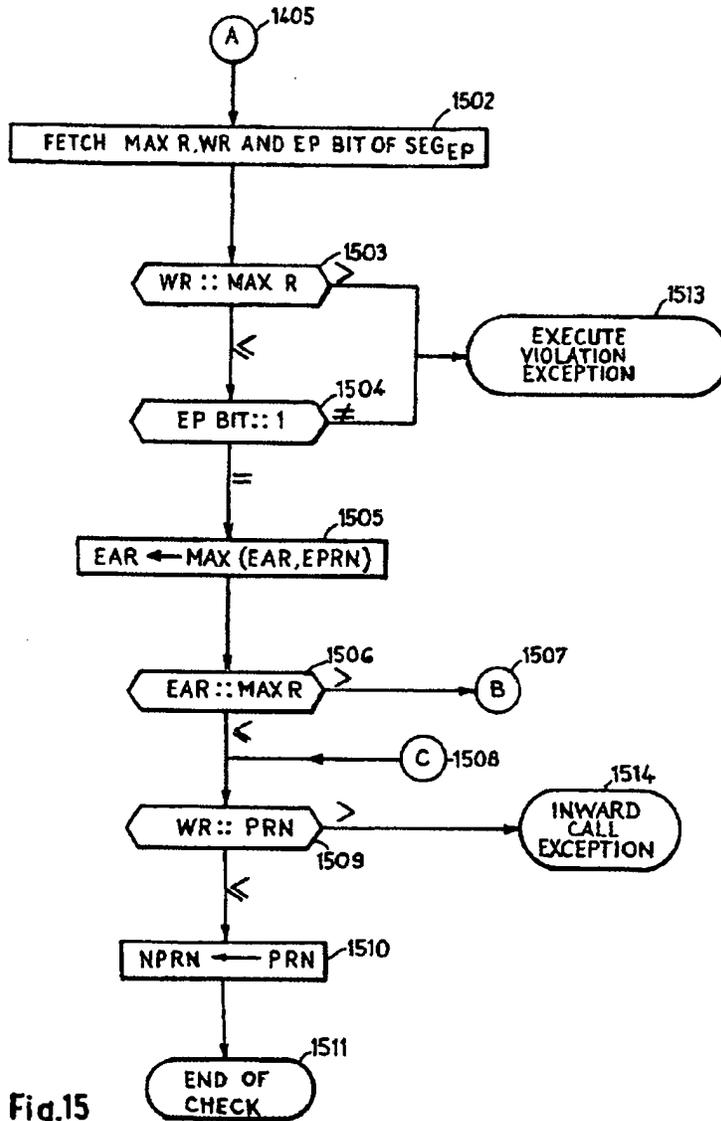


Fig.15

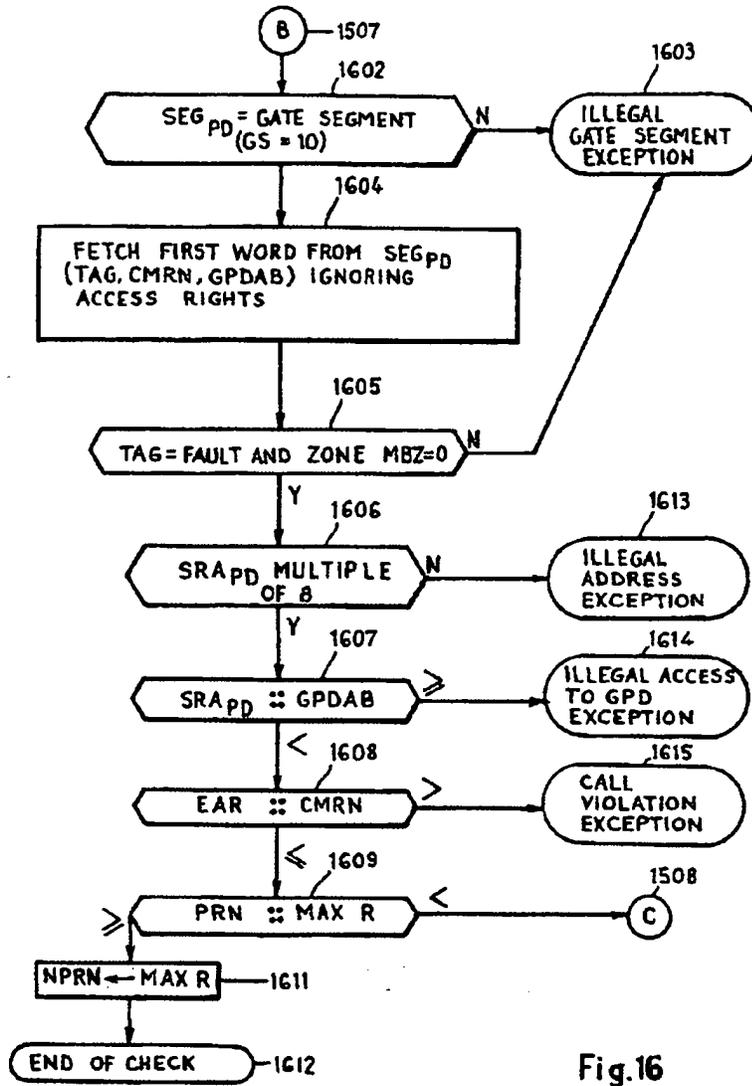


Fig.16

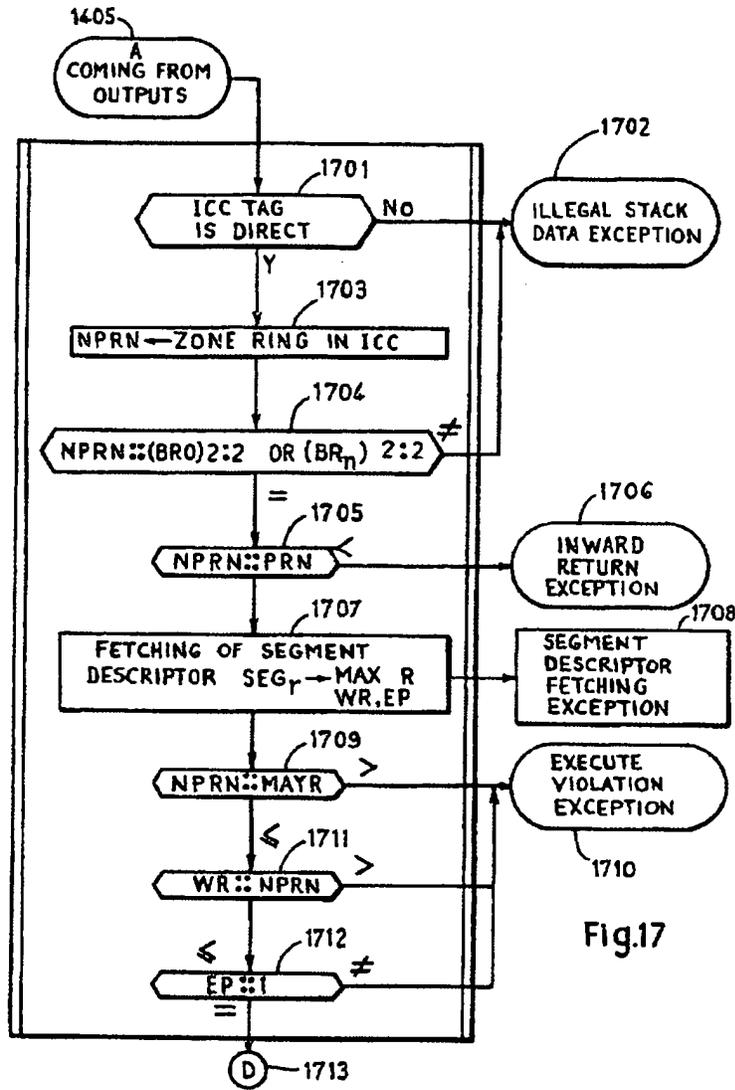


Fig.17

(12) UK Patent Application (19) GB (11) 2 236 604 A (13)

(43) Date of A publication 10.04.1991

(21) Application No 9009655.3

(22) Date of filing 30.04.1990

(30) Priority data
(31) 415984 (32) 02.10.1989 (33) US

(71) Applicant
Sun Microsystems Inc
(Incorporated in the USA - Delaware)
2550 Garcia Avenue, Mountain View, California 94043,
United States of America

(72) Inventor
John Richard Corbin

(74) Agent and/or Address for Service
Potts Kerr and Co
15 Hamilton Square, Birkenhead, Merseyside, L41 6BR,
United Kingdom

(51) INT CL^a
G06F 1/00

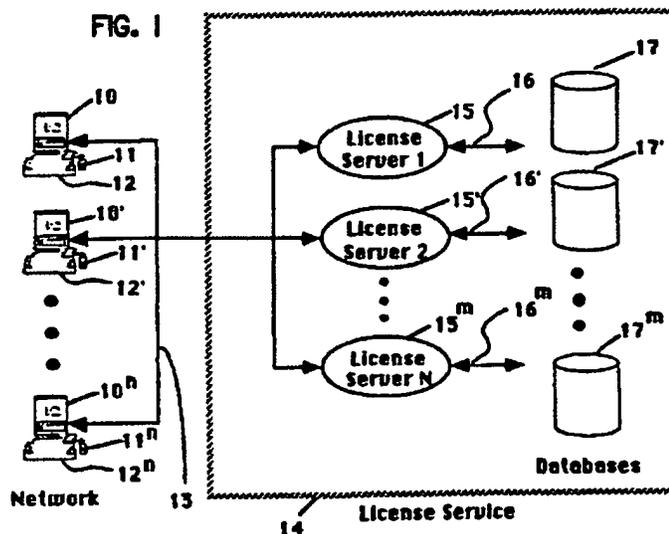
(52) UK CL (Edition K)
G4A AAP

(56) Documents cited
EP 0002390 A1 WO 88/02202 A1

(58) Field of search
UK CL (Edition K) G4A AAP
INT CL^a G06F 1/00 12/14
Online database: WPI

(54) Protecting against the unauthorised use of software in a computer network

(57) The present invention provides to a software application the verification and licence check out functions which are normally performed by a licence server. The encrypted licence information is contained in a licence token, and is stored in a database 17 controlled by the licence server 15. In contrast to the prior art where the server either grants or denies the request after verifying the user's credentials, the server in the preferred embodiment of the present invention finds the correct licence token for the software application and transmits the token to a licencing library. A licence access module attached to the application decodes the token. Routines in the licencing library coupled to the software application verify the licence information before issuing the licence and updating the token. The access module then encodes the updated token before returning it to the server. Because the verification and issuing function of a token are performed by a software application, the application rather than the server becomes the point of attack by unauthorised users. Reverse engineering the access module is less rewarding than attacking the server because the module reveals the contents of a small fraction of a database of licences.



At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

GB 2 236 604 A

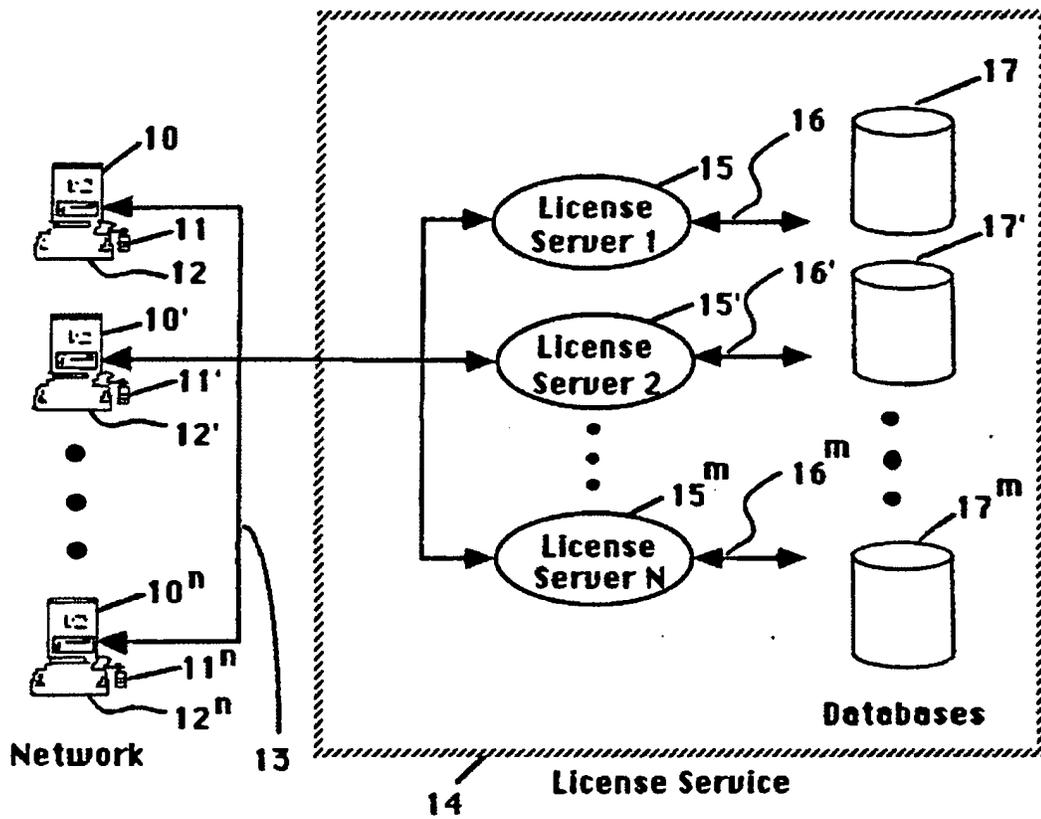


FIG. 1

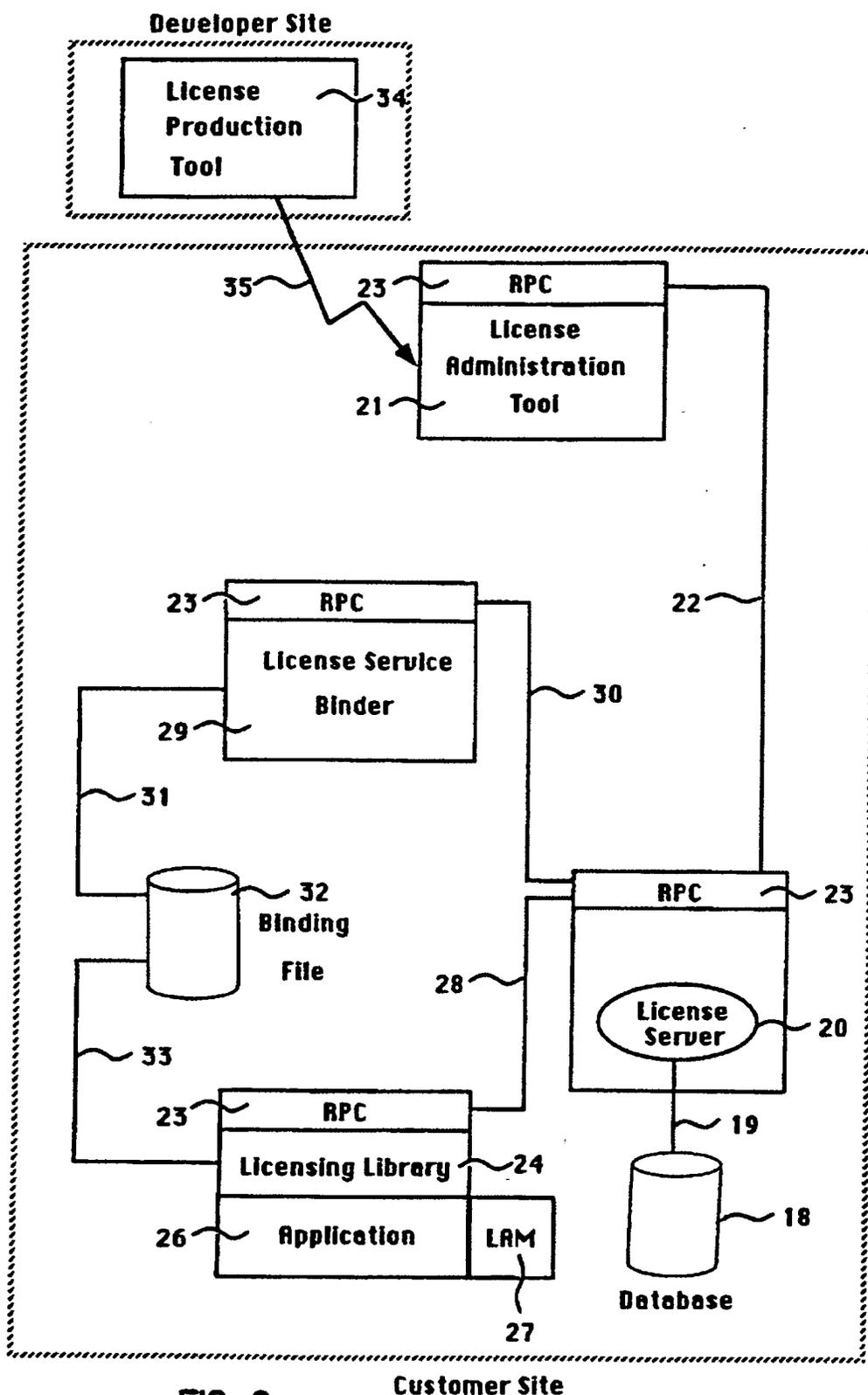


FIG. 2

Customer Site

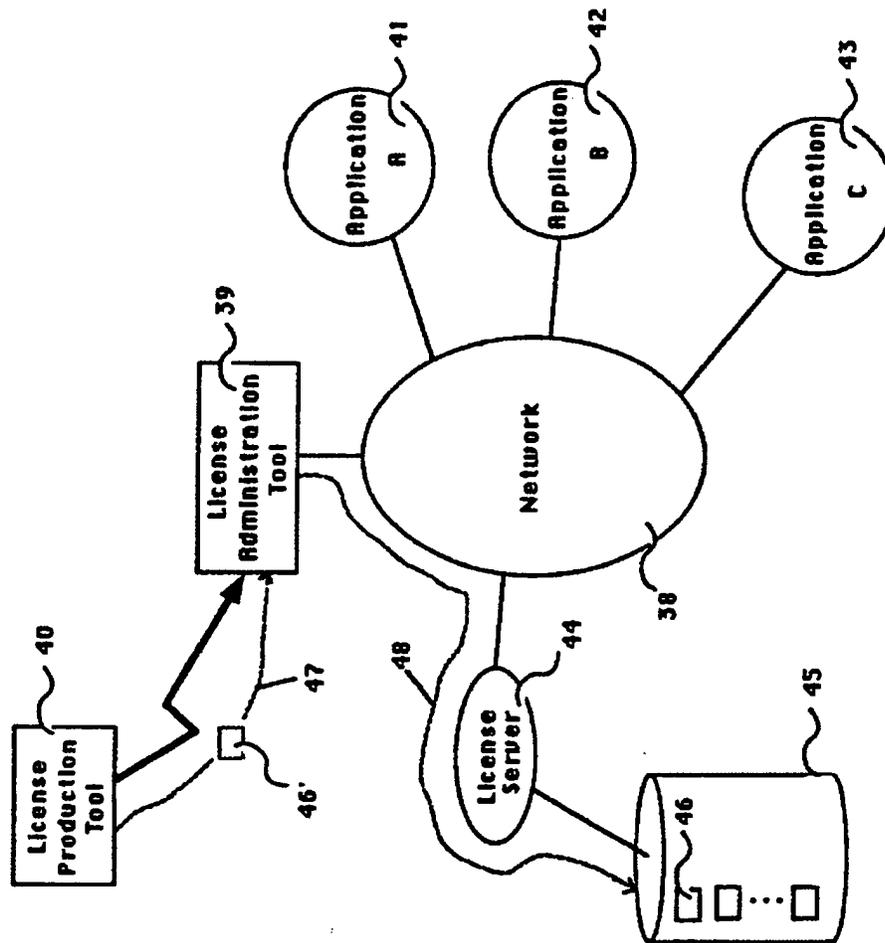


FIG. 3

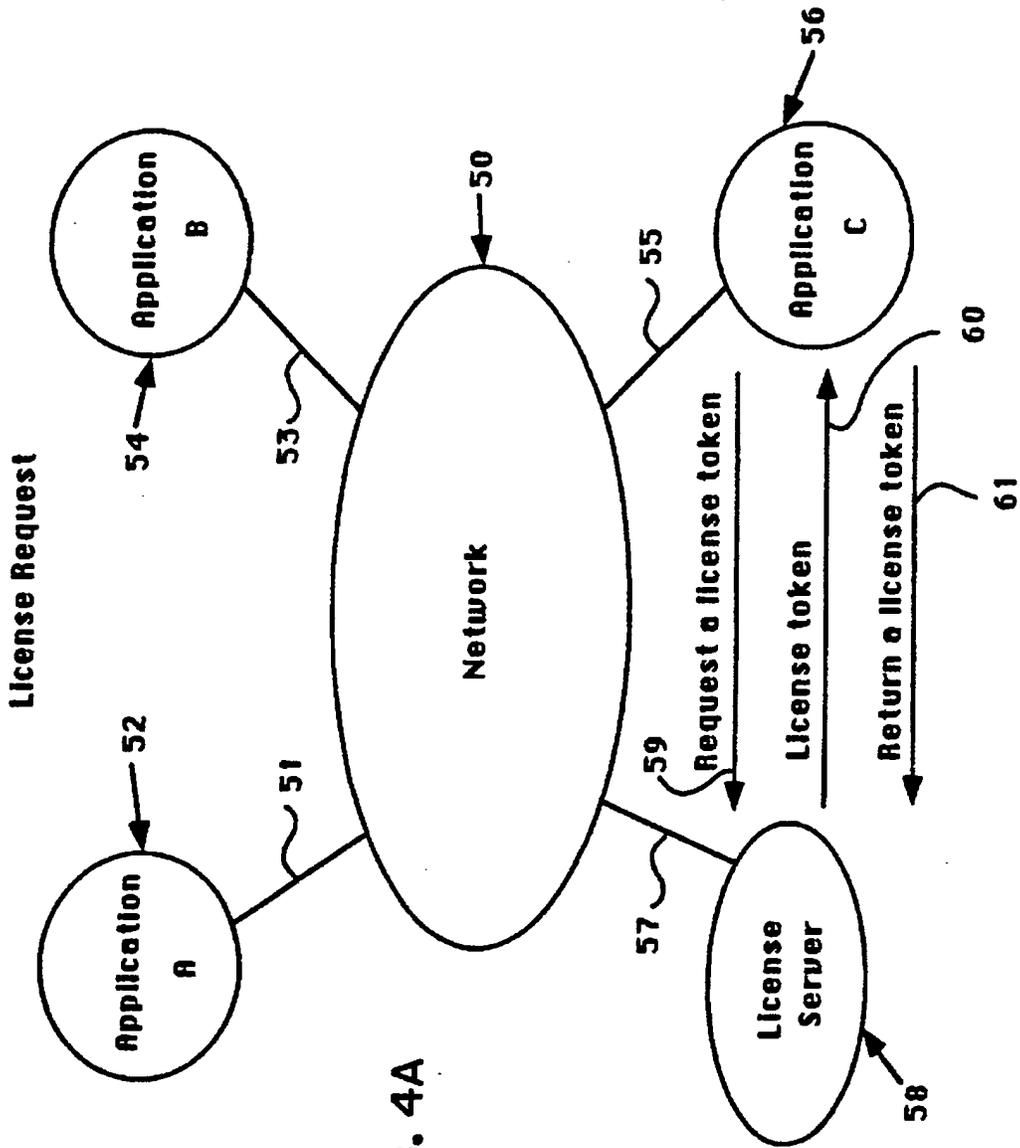


FIG. 4A

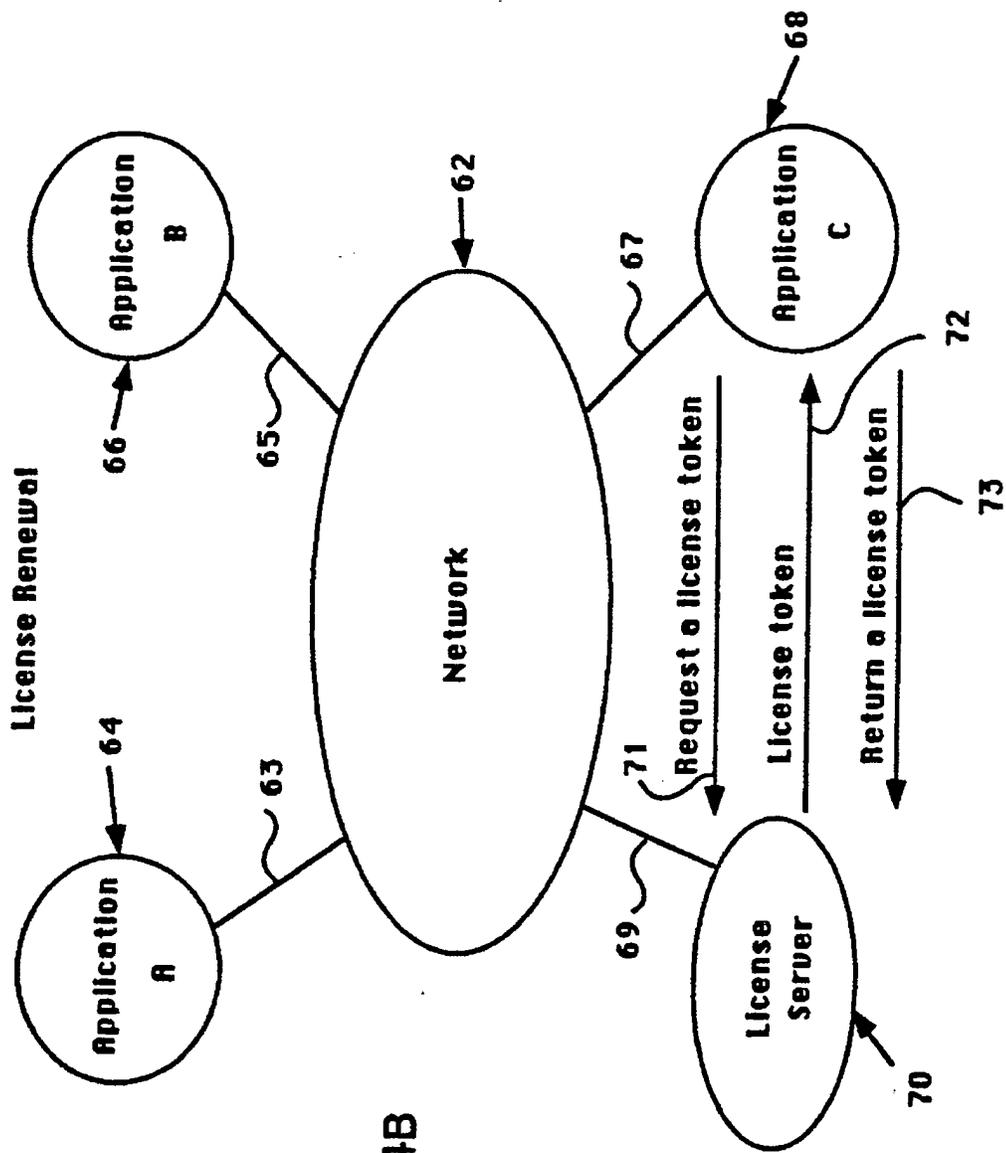


FIG. 4B

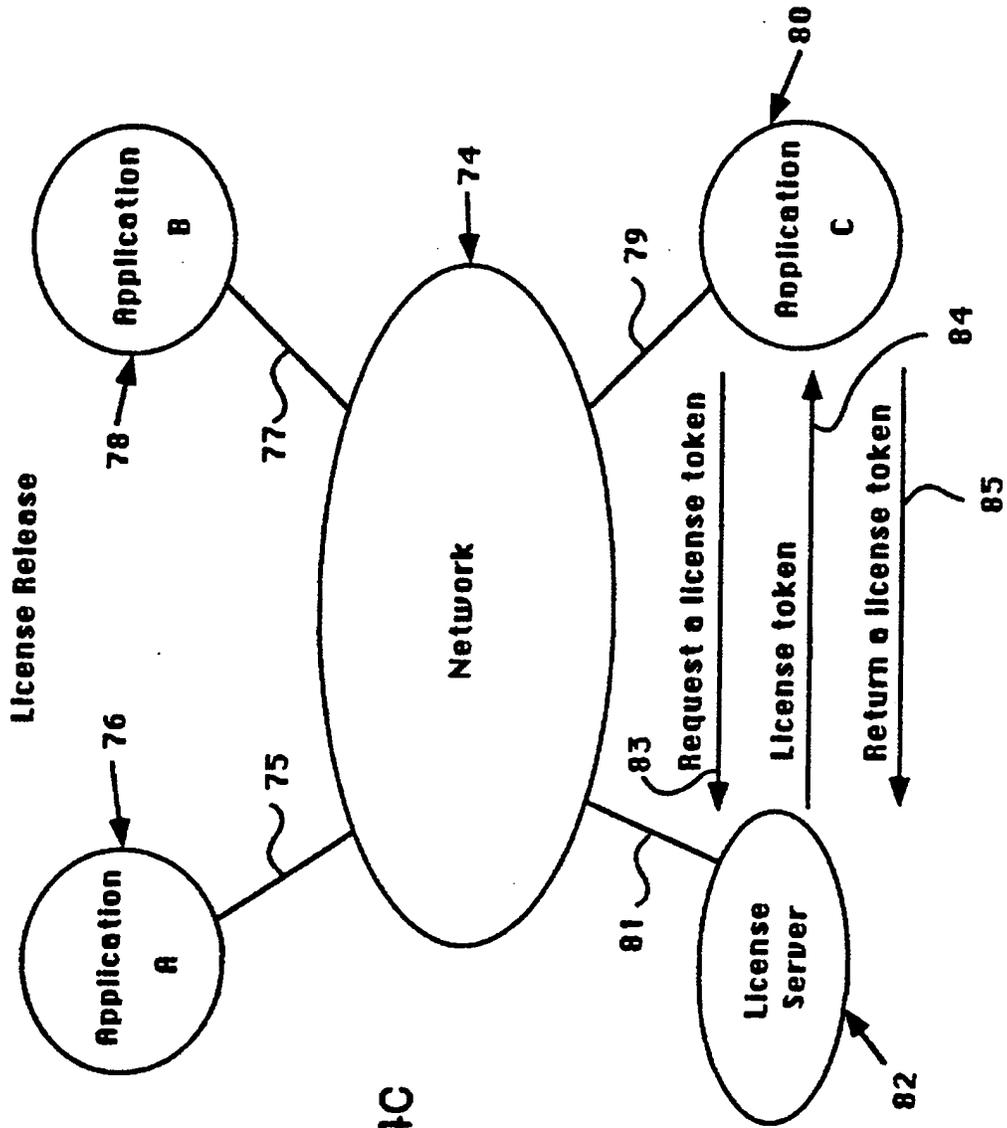


FIG. 4C

METHOD FOR PROTECTING AGAINST THE UNAUTHORIZED USE
OF SOFTWARE IN A COMPUTER NETWORK ENVIRONMENT

BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION

The present invention relates to a method for protecting against
5 the unauthorized use of a software application in a computer network
environment.

2. ART BACKGROUND

A computer network is typically an interconnection of machines or
10 agents over links or cables. The open access characteristics of a computer
network presents opportunities for the unauthorized copying of software, thus
eroding the licensing revenue potential of software developers. Traditionally,
either the entire network must be licensed (commonly referred to as a site
license), or each node where the software is run must be licensed (commonly
15 referred to as a node license). A node refers to a single machine, agent or
system in a computer network. A license is an authorization given by a
software developer to a customer to use a software application in a specific
manner.

20 A site license lets all users at a designated location or network
use the software application, regardless of their position on the network. This
flat-fee approach is an overkill for a low usage software application. A node
license not only ties a software application to a particular machine in a
network, but also is not cost effective for the infrequent use of a software
25 application. See, for example, U.S. Patent No. 4,688,169. Furthermore, if new
users of licensed nodes wish to use the software application, they are often
required to purchase additional licenses.

An alternative to a site license or a node license is the concept of
30 a concurrent usage license. A concurrent usage license restricts the number
of users allowed to use a software application at any given time, regardless of
their location on the network. Just as renters check out available copies of a

movie video from a video rental store, users on a network check out a software application from an agent on a first-come-first-serve basis. Thus, a concurrent usage license charges a fee for the use of a software application proportional to its actual use.

5

Methods to license a software application for concurrent use in a network environment are currently offered by Highland Software, Inc. and Apollo Computer, Inc. See, M. Olson and P. Levine, "Concurrent Access Licensing", *Unix Review*, September 1988, Vol. 6, No. 9. In general, the license for a software application is stored in a database controlled by a license server. A license server is a program that not only stores the license, but also verifies the user's credentials before checking out the license to the authenticated user. To protect against the authorized use, these methods to license concurrent usage rely on secured communications such as public/private key encryption. Under public/private key encryption, each user of the system has two keys, one of which is generally known to the public, and the other which is private. The private transformation using the private key is related to the public one using the public key but the private key cannot be computationally determined from the public key. See Denning, D., *Cryptography and Data Security*, Addison-Wesley, 1982. The encryption key is hidden in the license server to encrypt the database of licenses. Well designed public/private key encryption schemes are difficult to crack, especially if the license server is located in a trusted environment. A trusted environment is one whose access is limited to users having the proper credentials. However, a license server is more likely to be located at a customer's site and hence in an hostile environment. It follows that the license server is vulnerable to sophisticated intruders. Once the private key is decrypted, all sensitive information on the license server such as licenses are compromised.

30

It is therefore an object of the present invention to provide a more secure method to protect against the unauthorized use of software in a concurrent use licensing environment.

SUMMARY OF THE INVENTION

The present invention provides to the software application the verification and license check out functions which are normally performed by a license server. The preferred embodiment of the present invention comprises a computer network including a plurality of agents running at least one license server and at least one software application. The license server controls a database of an agent containing the license information for the software application. The license information is contained in a license token, and is stored in the database controlled by the license server. The license token is a special bit pattern or packet which is encrypted by the software vendor of the application software. The software application communicates with the license server through a licensing library. The licensing library is a collection of library routines that the software application invokes to request or renew a license from the license server. Before a software application obtains a license, the license token must be decoded by a license access module. The license access module, which is linked with the software application and the licensing library is a program that decodes the license token from a vendor specific format to a licensing library format.

When an user wishes to run a software application, the licensing library invokes a call to request a license token from the license server. In contrast to the prior art where the license server either grants or denies the request after verifying the user's credentials, the license server in the preferred embodiment of the present invention finds the correct license token for the software application and transmits the license token to the licensing library. The license access module attached to the licensing library decodes the licensing token. Routines in the licensing library coupled to the software application verify the license information before checking out the license and updating the license token. The license access module encodes the updated license token before returning it to the license server.

Because the verification and check out function of a license token are performed by a software application, the software application rather than the license server becomes the point of attack by unauthorized users. Reverse engineering the license access module is less rewarding than attacking the

5 license server because the license access module reveals the contents of a fraction of a database of licenses. By the time most attackers crack the license access module, the software vendors would most likely introduce newer versions of the software application and new license access modules for them. Thus the present invention provides a more secure method for protecting

10 against the unauthorized use of a software application in a computer network environment without modifying the underlying computer network.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a network environment employing the present invention.

5

Figure 2 describes the architecture of a network licensing scheme employing the preferred embodiment of the present invention.

Figure 3 describes the installation of a license token in the preferred embodiment of the present invention.

10

Figure 4a illustrates the use of a license token to request a license from a license server in the preferred embodiment of the present invention.

15

Figure 4b illustrates the use of a license token to renew a license from a license server in the preferred embodiment of the present invention.

Figure 4c illustrates the use of a license token to release a license from a license server in the preferred embodiment of the present invention.

20

NOTATION AND NOMENCLATURE

The detailed description that follows is presented largely in terms of algorithms and symbolic representations of operations on data bits and data
5 structures within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art.

10 An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. These steps are those requiring physical manipulation of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It proves
15 convenient at times, principally for reasons of common usage, to refer to these signals as bit patterns, values, elements, symbols, characters, data packages, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

20 Further, the manipulations performed are often referred to in terms, such as adding or comparing, that are commonly associated with mental operations performed by a human operator. No such capability of a human operator is necessary, or desirable in most cases, in any of the operations described
25 herein that form part of the present invention; the operations are machine operations. Useful machines for performing the operations of the present invention include general purpose digital computers or other similar devices. In all cases there should be borne in mind the distinction between the method of operations in operating a computer and the method of computation itself. The
30 present invention relates to method steps for operating a computer in processing electrical or other (e.g. mechanical, chemical) physical signals to generate other desired physical signals.

The present invention also relates to an apparatus for performing these operations. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer as selectively
5 activated or reconfigured by a computer program stored in the computer. The algorithms presented herein are not inherently related to any particular computer or other apparatus. In particular, various general purpose machines may be used with programs written in accordance with the teachings herein, or it may prove more convenient to construct a more specialized apparatus to
10 perform the required method steps. The required structure for a variety of these machines will appear from the description given below.

DETAILED DESCRIPTION OF THE INVENTION

The following detailed description is divided into several sections. The first of these sections describes a general network environment for accessing a database of licensed software programs. Subsequent sections discuss the details of a method for protecting against the unauthorized use of a software application.

I. General Network Environment

Referring to Figure 1, computer network environment comprises a plurality of data processing devices identified generally by numerals 10 through 10ⁿ (illustrated as 10, 10' and 10ⁿ). These data processing devices may include terminals, personal computers, workstations, minicomputer, mainframes and even supercomputers. For the purposes of this Specification, all data processing devices which are coupled to the present invention's network are collectively referred to as "agents". It should be understood that the agents may be manufactured by different vendors and may also use different operating systems such as MS-DOS, UNIX, OS/2, MAC OS and others. Particular examples of suitable agents include machines manufactured by Sun Microsystems, Inc., Mountain View, Calif. Each of the agents has an input device such as a keyboard 11, 11' and 11ⁿ or a mouse 12, 12' and 12ⁿ. As shown, agents 10 through 10ⁿ (illustrated as 10, 10' and 10ⁿ) are interconnected for data transfer to one another by a common cable 13. It will be appreciated by one skilled in the art that the common cable 13 may comprise any shared media, such as coaxial cable, fiber optics, radio channel and the like. Furthermore, the network resulting from the interconnection of the cable 13 and agents 10 through 10ⁿ (illustrated as 10, 10' and 10ⁿ) may assume a variety of topologies, such as ring, star, bus, and may also include a collection of smaller networks linked by gateways or bridges.

Referring again to **Figure 1** is a license service 14. The license service 14 is a resource shared by every agent connected to the network. In the preferred embodiment of the present invention, the license service 14 comprises license servers 15 through 15^m (illustrated as 15, 15' and 15^m) and databases 17 through 17^m (illustrated as 17, 17' and 17^m), where m is less than or equal to n. A license server is a program that runs on an agent with a memory storage capability. Each license server 15 (illustrated as 15, 15' and 15^m) communicates with a database 17 stored in memory on the agent over an interface 16 (illustrated as 16, 16' and 16^m). As will be described in detail below, the database 17 stores licensing information for various software applications which are purchased and authorized to run in the computer network environment. The license server is not limited to run on a specific agent, but can operate on any agent including the agent on which the user is to operate the application. Thus, any agent connected to the network may function as a license server as well as a device on which a user may operate application software. As will be described below, the license server does not perform verification of licenses of application software; rather the license server is passive and provides storing, locking, logging, and crash recovering function for the application software.

20

Figure 2 illustrates the architecture of a network licensing scheme of the present invention. The architecture comprises a database 18, database interface 19, license server 20, licensing library 24, License access module 27, license administration tool 21, license service binder 29, and license production tool 34.

25

The database 18 stores licensing information and application usage data. Preferably the database 18 comprises a plurality of records which contain the following information:

	<u>Database Element</u>	<u>Description</u>
	Unique Key Table	Keys for all other tables
	Vendor Table	Vendor's ID and name
	Product Table	Product number and name
5	Version Table	Version number and date
	License Table	License #, exp date, total units
	License Token Table	Stores encoded license token
	Unit Group Table	A group's allocation of license
	Group List Table	Name of the group
10	Allowed Users Table	Credentials of allowed users
	Current License Use Table	Applications using a license
	Lock Table	Locked records in database
	Authorized administrator Table	Login names of administrators
	License Operation Log Table	Administrator's log information
15	License Usage Log Table	Request handle plus Client Log
	License Queue Log Table	License wait queue
	Application Message Log Table	Application specific messages

20

A database interface 19 provides communication between the license server 20 and the database 18 in order to prevent concurrent access to the same database record by multiple users which can cause the data in the record to become corrupted. Thus, only the owner of the lock can read from and write to the locked record during the usage of the application.

The license server 20 operates on an agent and interfaces the database 18 to license administration tool 21, licensing library 24 and license service binder 29. The license server 20 communicates with the license administration tool 21, licensing library 24 and license service binder 29 via an interface 23. Preferably the interface 23 is a remote procedure call

mechanism which permits a process operating on one device or agent
connected to the network to request a resource or service from a remote device
or agent connected to the network. See A. Birrell and B. Nelson, "Implementing
Remote Procedure Calls," *ACM Transaction on Computer Systems*, February
5 1984, Vol. 2, No. 1.

Multiple license servers may reside on multiple agents. Preferably the
license server 20 operates in a background mode of the agent such that its
operation is transparent to a user of that agent. More particularly, as will be
10 described below, the license server 20 provides the following functions: (1)
servicing the requests from the licensing library 24 for license token; (2)
maintaining a wait queue for requests to the database 18 when no licensing
units are available; (3) generating locks for exclusive access to database
18; and (4) providing access to information in the database 18.

15 The licensing library 24 is a set of library routines which enable the
application 26 to request licensing service from the license server 20. Upon
receiving the request for service from the licensing library 24, the license
server 20 retrieves a license token from the database 18 and transmits it to the
20 licensing library 24. The licensing library 24 is linked with the application 26
and communicates with the license server 20 over a path 28 with, preferably,
a remote procedure call mechanism 23. Among the major library calls in the
licensing library 24 is the application's request for a license from the license
server 20. Other important library calls include the request to renew and to
25 release a license. The use of the license token to accomplish the request for
the various licensing service will be described in detail below.

The license access module (LAM) 27 is prepared by the software
vendor 24 to decode the license token. Once decoded, the application 26 via
30 routines in the licensing library verifies the licensing information in the license
token and determines whether a license may be checked out. The LAM 27

also encodes the license token before the application returns it to the database 18 via license server 20. The license access module 27 is described in further detail below.

5 The license administration tool 21 is utilized by the network administrator to perform administrative functions relevant to the concurrent usage of a software application. The license administration tool 21 may run on any agent connected to the computer network. The license administration tool 21 is primarily used to install the license token into the database 18 through the
10 license server 20. The functionality of the license administration tool 21 includes: (1) starting or terminating a license server, (2) accessing a database controlled by a license server; and (3) generating and printing reports on license usage.

15 The application 26 may not access the database 18 directly; rather, the request for a license is made through the licensing library 24 to the license server 20 over a path 28. Most network licensing schemes employ secured communication between the licensing library 24 and the license server 20. In contrast, the present invention uses the license access module (LAM) 27 the
20 license library 24 and a plurality of license tokens to protect against the unauthorized use of software application in a computer network.

 Referring once again to Figure 2, a license service binder 29 is shown coupled to the license server 20 over a path 30. The license service binder
25 29 is invoked by means known in the art, such as a network service program. The license service binder 29 locates all agents that are designated as servers on the network, and keeps track of which server is servicing which application. The license service binder 29 contacts each server on its table of available servers and requests a list of products it serves. Finally the license service
30 binder 29 writes the contents of the table of available license servers and the list of products into a binding file 32 over a path 31. In Figure 2, the binding file 32 is coupled to the licensing library 24 over a path 33. The application 26

queries the binding file 32 to see which license server can service its request for a license.

A license production tool 34 is used by the software vendor to create a
5 license token for transmittal to the network administrator. Receiving the license token, the network administrator installs it with the license administration tool 21 into the database 18 through license server 20.

II. License Token

10 Referring to Figure 3, the creation of a licensé token in a computer network employing the preferred embodiment of the present invention will be described. A computer network 38 is shown coupled with a license administration tool 39 and a single license server 44. The license server 44 communicates with a database 45. Applications 41, 42, and 43 are shown
15 requesting licensing service from the license server 44. When a customer purchases a license for an application, such as a CAD/CAM program for its research and development department, the software vendor creates a license token with a license production tool, and delivers the license token to the customer's network administrator. A license token is a special bit pattern or
20 packet representing a license to use a software application. The network administrator installs the license token 46 into the database of the license server using the license administration tool 39. Unlike the token used in a token ring which is passed from agent to agent, a license token in the preferred embodiment of the present invention is passed only between a license server
25 and a licensing library for a predetermined amount of time. The predetermined amount of time corresponds to the time the license token is checked out of the license server. Currently, the license token is checked out to an application for no more than ten seconds, and the license token is returned as quickly as possible to the issuing license server. The license token 46 contains
30 information encrypted in the vendor's format such as vendor identification, product and version numbers as well as the number of license units purchased

for the license token. A license unit corresponds to the license weighting for an agent connected to the computer network. For example, powerful workstations could require more license units to use a software application than an average personal computer.

5

The software vendor produces a license token using a license production tool 40. A path 47 illustrates how a license token 46' makes its way to a license administration tool 39 at the customer's site. There, the system administrator installs the license token 46' as license token 46 into the license database 45 of the license server 44. A path 48 indicates the transfer of the license token 46' from the license administration tool 39 to the license server 44 and into the database 45 as license token 46. The license server 44 is now ready to entertain requests from applications 41, 42, and 43 for a license to use the application corresponding to token 46 as well as other applications represented in its database 45.

It should be understood that each network may have a plurality of license servers and each license server may have in its database a plurality of license tokens for a variety of software applications. Referring again to Figure 3, if application A 41 requests and checks out the license token 46 for less than ten seconds, applications B and C 42, 43 would be unable to check out the license token 46 if their requests were made during the same time application 41 is checking out a license from the license token 46 because of the locking mechanism provided by database interface 19. Thus, to achieve concurrent license usage in network 38, it is preferred that the network administrator installs more than one license server. To minimize the task of recovering from license server crashes, it is also preferred that the system administrator spreads the license units for any one application among a plurality of strategically located license servers. For instance, if a network has four license servers, the network administrator may want to allocate the twenty license units for a particular popular application among four license tokens with

same access to any agent in a network, including the license server. The security of the licensing scheme can be compromised by a user who decrypts the license server's private key. Once the unauthorized user determines the server's private key, he can decrypt all sensitive information on the license server. Should all license servers use the same key, as is frequently done, then all the security of the applications served by all the license servers will be compromised.

The license access module 27 first translates a license token from a vendor specific format to a format usable by the licensing library 24. The license access module accomplishes the translation in two modules. One module translates or decodes a license token from a vendor specific format to a licensing library format. The second module translates or encodes the updated license token from the licensing library format to the vendor specific format. The second module is invoked anytime the licensing library updates the information in a license token.

Upon receiving the license token in the licensing library format, the licensing library invokes routines which verify the correctness of the license by reviewing the following license information stored in the token: (1) flag, (2) maintenance contract date, (3) host name and domain, (4) product name, (5) host id number, (6) license serial number, and (7) expiration date of license. This is compared to the information maintained by the application. If the information matches, the license is verified. After completing the verification process, a routine in the licensing library is initiated which checks out the license by decrementing the license units in license token by the number of licensing units being checked out.

The decoding and encoding routines allow software vendors to implement their own security mechanism to protect their licenses from unauthorized use even though they reside at the customer's site.

Below is an example of a sample application using the licensing library and the license access module written in C language:

```

5  #define LIC_RENEWAL_TIME (60)           /set renewal time for this session/
   #define EST_LIC_RENEWAL_TIME (LIC_RENEWAL_TIME x .9)

   NL_vendor_id NL_Vendor_id = 1223;     /set vendor #/
   NL_prod_num NL_Prod_num = "02"       /set product #/
10  NL_version NL_Version = ( 12/20/88, "1.0" ); /set version id #/

   --
   status = NL_init (vendor_id, NULL, &job_id); /initialize license service/
   if (status != NL_NO_ERROR) /accept job id if no error/
   {
15     fprintf (stderr, "nl_init failed - error =
        %d\n", status); /error message if error and
                           return/

        return;
   }

20  units = 3;
   code_funcs.encode_p = nl_encode; /pointer to encode function/
   code_funcs.decode_p = nl_decode; /pointer to decode function/
   if (signal (SIGALRM), alarm_intr ) == (void *) -1) /set alarm if no
                                                       error/

25     {
        perror ("Cannot set SIGALRM"); /otherwise, error message/
        return;
   }

   status = NL_request (job_id, NL_Prod_num, /request a license/
30   &NL_Version,
   units, LIC_RENEWAL_TIME, NL_L2_SRCH,
   &code_funcs, NULL,
   &req_handle, NULL, &app_info);

   if (status != NL_NO_ERROR) /no error, license checked
35   {
        fprintf (stderr, "nl_request failed - error =
        %d\n", status); /otherwise, error message/
        return;
   }

40   /*
   * We got a license /license request successful/
   */

   alarm (EST_LIC_RENEWAL_TIME); /set alarm for license renewal
45   time/
   Application Runs /runs application/

   --
   status = NL_release (req_handle); /request to release a license/
   if (status != NL_NO_ERROR)
50   {
        fprintf (stderr, "nl_release failed - error = /otherwise, error

```

```

        %d\n", status);           messages/
        return;
    }

5   int
    alarm_intr ()
    {
        status = NL_confirm (req_handle,    /renew licensing unit with
        LIC_RENEWAL_TIME, NULL);          licensing server/
10   /*
        * Verify vendor private information
        */
    }

    If (status != NL_NO_ERROR)
15   fprintf (stderr, "nl_confirm failed - error =    /otherwise, error
        %d\n", status);              message/
        {
            puts ("license renewed")    /successful license
        }                             renewal/
20

```

The sample application given above is accompanied by self-explanatory annotation to the right margin of the codes. Of particular interest are code_func.encode_p and code_func.decode_p. Encode_p and decode_p are pointers to the software vendor's encode and decode routines, respectively. Taking the pointers in the code_func variable, the licensing library can use the pointers to invoke the decoding and encoding routines in the license access module. The three major licensing library routines, request for a license (NL_request), release a license (NL_release) and renew a license (NL_confirm) invoke the decoding and encoding routines. For example of a license access module, see Appendix 1.

In implementing the license access module, the license server becomes merely a repository for license tokens. The licensing library coupled to the application performs the procedure of authenticating the license token prior to granting a license and therefore access to run the application.

Because the level of security of the system is dictated by the license access module, the software vendors are free to make the license access module as simple or as complex as they desire. In particular, they are free to

adopt any of the encryption schemes as part of their encryption routines. If the security mechanism is broken, and the encryption known to others, then the software vendors can easily remedy the situation by releasing a new version of the product with a new license access module.

5

While the present invention has been particularly described with reference to Figures 1-4 as well as Appendix 1, and with emphasis on certain language in implementing a method to protect against the unauthorized use of software application in a computer network environment, it should be

10 understood that they are for illustration only and should not be taken as limitation upon the invention. In addition, it is clear that the method of the present invention has utility in any application run in a computer network environment. It is contemplated that many changes and modifications may be

15 the invention disclosed above.

CLAIMS

1. In a computer network environment including a plurality of software applications licensed to run on at least one network of agents, said applications located on said agents wherein use of the application on a particular agent is permitted upon the grant of a license, said license being requested by a user from said agent of said applications, a system for protecting against the unauthorized use of said applications comprising:

license token means for storing licensing information of said applications; license server means connected to said agents for communicating with said applications, said license server means having a database which stores said license token means, said license server means further retrieving said license token means from said database upon a request for a license by said applications, said license server means further transmitting said license token means to said applications;

license access means connected to said agents for decoding and encoding said license token means from said license server means, said license access means being integrated with said applications, said license access means receiving said license token means from said license server means; and

licensing library means connected to said agents for verifying said decoded license token means before access to said license is granted, said licensing library means being integrated with said applications.

2. The system as defined in claim 1, wherein each said license token means containing licensing information for at least one version of each said applications.

3. The system as defined in claim 1, wherein the contents of said license token means is encrypted.

4. The system as defined in claim 1, wherein said license token means is passed between said license server means and said licensing library means for a predetermined time period.

5. The license token means as defined in claim 4, wherein during said predetermined time period, only one said applications may check out one said license token means.

6. The system as defined in claim 1, wherein said license server means receives said request for a license from said applications, said license server searches in said database for a license token means storing the license requested by said application before retrieving said license token means.

7. The system as defined in claim 1, wherein said license access means decodes the contents of said license token means before said licensing library means verifies said license token means.

8. The system as defined in claim 1, wherein said license access means encodes said license token means after said licensing library verifies said license token means and prior to returning said license token means to said license server means.

9. The system as defined in claim 1, wherein said licensing library verifies said license token means by

comparing the licensing information stored in said license token means with the licensing information maintained by said application.

10. The system as defined in claim 1, wherein said licensing library means checks out said license of said application in response to a positive comparison of the license information.

11. The licensing library means as defined in claim 10, wherein said license for said application being checked out after said licensing library verifies said license token means.

12. In a computer network environment including a plurality of software applications licensed to run on at least one network of agents, said applications located on said agents wherein use of the application on a particular agent is permitted upon the grant of a license, said license being requested by a user from said agent of said applications, a system for protecting against the unauthorized use of said applications comprising:

license token means for storing licensing information of said applications;

license server means connected to said agents for communicating with said applications, said license server means having a database which stores said license token means, said license server means further retrieving said license token means from said database upon a request for a license by said applications, said license server means further transmitting said license token means to said applications;

license access means connected to said application and accessible from said agents for decoding and encoding said license token means from said license server means, said license access means being integrated with said applications;

licensing library means connected to said application and accessible from said agents for verifying said decoded license token means before access to said license is granted, said licensing library means being integrated with said applications; and

license binding means connected to said license server means and to said licensing library means for constructing a binding file, said binding file informing said licensing library means which of said license server means may grant a license to said application.

13. The system as defined in claim 12, wherein said licensing library means are located on the same agents as said applications.

14. The system as defined in claim 12, wherein said license sever means are located on the same agents as said licensing library means.

15. The system as defined in claim 12, wherein each said license token means contains licensing information for at least one version of each of said applications.

16. The system as defined in claim 12, wherein the contents of said license means is encrypted.

17. The system as defined in claim 12, wherein said license token means is passed between said license server

means and said licensing library means for a predetermined time period.

18. The license token means as defined in claim 17, wherein, during said predetermined time period, only one of said applications may check out one said license token means.

19. The system as defined in claim 12, wherein said license server means further transmit said license token means to said licensing library means.

20. The system as defined in claim 12, wherein said license access means decodes the contents of said license token means before said licensing library means verifies said license token means.

21. The system as defined in claim 12, wherein said license access means encodes said license token means after said licensing library verifies said license token means and prior to returning said license token means to said license server means.

22. The system as defined in claim 12, wherein said license binding means constructs said binding file by contracting each said license server means to request for a list of applications it serves, said binding file containing said list of applications available from said license server means.

23. In a computer network environment including a plurality of software applications licensed to run on at least one network of agents, said applications located on

said agents wherein use of the application on a particular agent is permitted upon the grant of a license, said license being requested by a user from said agent of said applications, a system for protecting against the unauthorized use of said applications substantially as hereinbefore described with reference to the accompanying drawings.

(12) UK Patent Application (19) GB (11) 2 309 364 (13) A

(43) Date of A Publication 23.07.1997

(21) Application No 9700921.1
 (22) Date of Filing 17.01.1997
 (30) Priority Data
 (31) 06588848 (32) 19.01.1996 (33) US

(71) Applicant(s)
 Northern Telecom Limited
 (Incorporated in Canada - Quebec)
 World Trade Center Of Montreal,
 380 St Antoine Street West, 8th Floor, Montreal,
 Quebec H2Y 3Y4, Canada

(72) Inventor(s)
 David Allan
 Liam Casey
 Adrian Jones

(74) Agent and/or Address for Service
 M C Dennis
 Nortel Patents, London Road, HARLOW, Essex,
 CM17 9NA, United Kingdom

(51) INT CL⁶
 H04L 9/30

(52) UK CL (Edition O)
 H4P PDCSC
 U1S S2204 S2208 S2209

(56) Documents Cited
 EP 0328232 A2 WO 95/23468 A1

(58) Field of Search
 UK CL (Edition O) H4P PDCSA PDCSC
 INT CL⁶ H04L 9/30 9/32
 Online:WPLINSPEC

(54) Public/private key encryption/decryption

(57) In a hybrid fiber-coax distribution network, communications between a central station and particular end stations are encrypted using a working key (WK) of a symmetric encryption scheme. The central station has a public and private key (PPK) of a PPK encryption scheme, and some of the end stations can also each have a respective PPK. To provide secure communications for each end station, if the end station has a PPK, then the respective WK is generated in the central station and communicated, encrypted using the end station's public key (PK), to the end station. Otherwise, the WK is generated in the end station and communicated, encrypted using the central station's PK, to the central station. An individual identifier for each end station, and a cryptographic signature at least for end stations not having a PPK, can be communicated to the central station for authentication of the end stations.

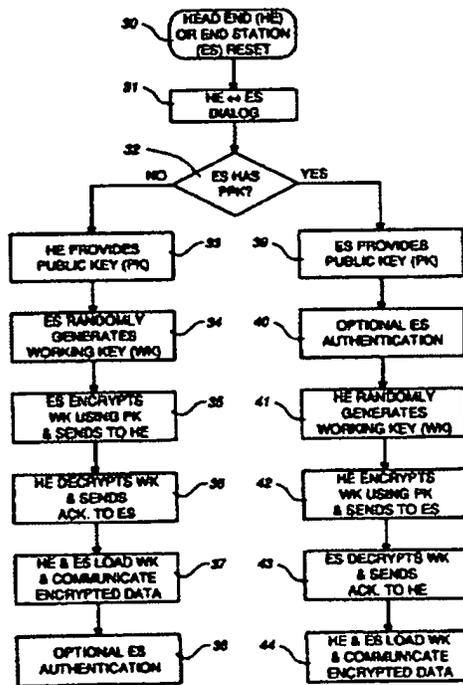


Fig. 2

GB 2 309 364 A

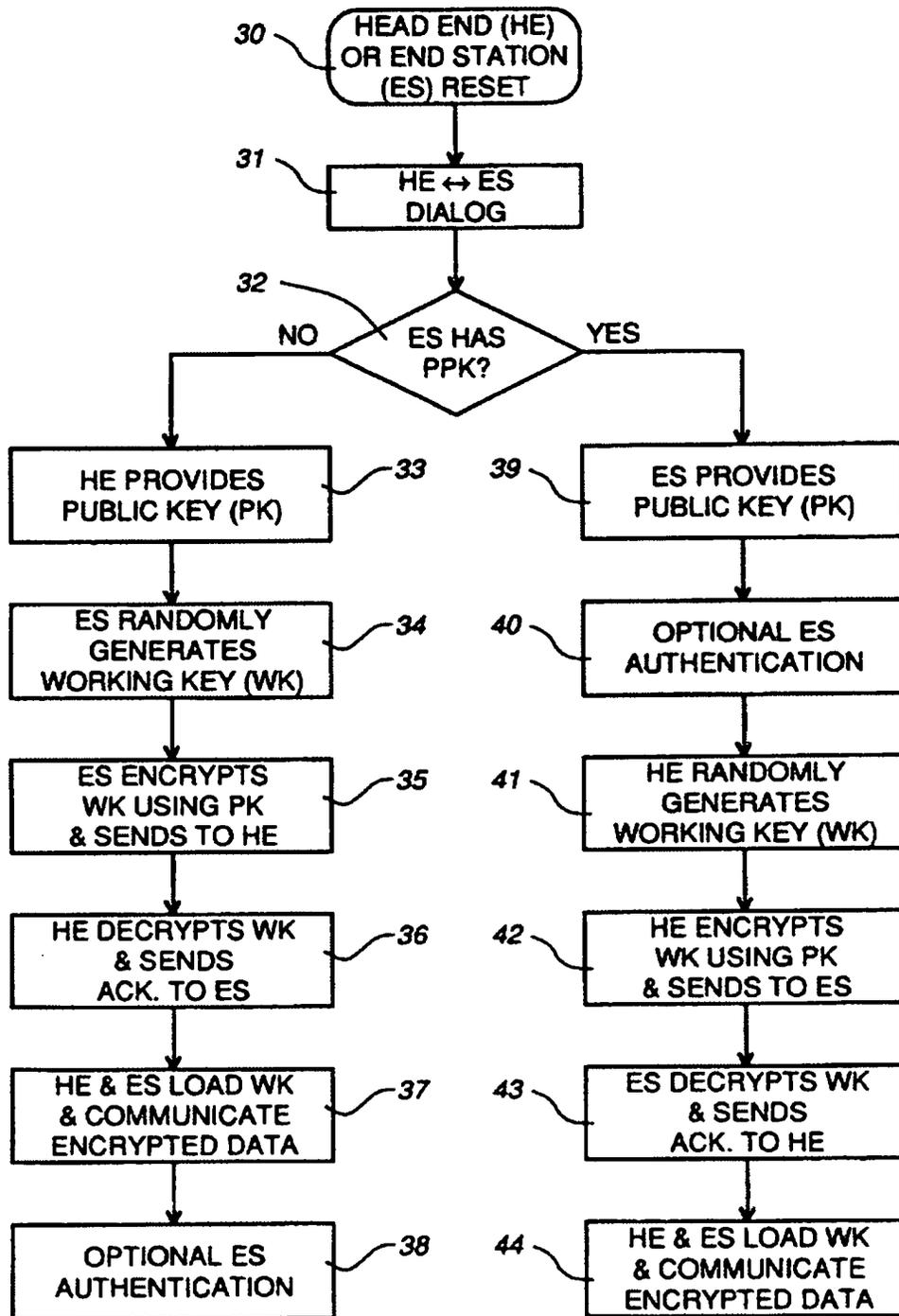


Fig. 2

FACILITATING SECURE COMMUNICATIONS
IN A DISTRIBUTION NETWORK

This invention relates to methods of facilitating secure communications in a distribution network, such as for example a coaxial cable or hybrid fiber-coax (HFC) network.

Background of the Invention

A distribution network, such as an HFC network in which data is communicated to subscriber end stations via optical fiber and coaxial distribution cables, is a point-to-multipoint network in which data addressed to and intended for any particular subscriber is also inevitably supplied via the network to other subscribers. If the data is not scrambled or encrypted, it can be easily monitored by these other subscribers, leading to a loss of subscriber privacy and a loss of revenues for data suppliers when the data (e.g. television programs) is supplied for a fee. Accordingly, it is important to provide a desired level of security in the data communications in a distribution network.

While various encryption and decryption schemes are known, these have a number of disadvantages associated with them in the environment of a distribution network. A significant factor in this respect is the cost and security of subscriber end stations. As a distribution network will contain large numbers of subscriber end stations, it is commercially necessary that the cost of each end station be kept relatively low. It is therefore desirable to avoid incorporating expensive security schemes in the subscriber end stations. However, subscriber end stations are also easily subject to theft, tampering, and duplication, so that complicated schemes have been considered necessary to provide adequate security.

For example, a security scheme can be implemented using an encryption key which can be stored in the subscriber end station. To prevent access to the encryption key, the store in the subscriber end station, and data lines to and from this store, must also be made physically secure. This leads to extra complexity and costs. Different subscribers may have differing security and privacy needs, which makes it desirable for the network to accommodate differing security schemes and end station costs.

A further security-related desirable aspect of a distribution network is an ability for authentication of subscriber end stations, typically using a unique end station identity which can be physically incorporated (e.g. hard wired) into the end station during manufacture.

Encryption schemes can be divided into those involving public and private keys (PPK) and those involving symmetric keys. In PPK schemes, a first station can distribute its public key, in accordance with which a second station can encrypt data and send the encrypted data to the first station, which decrypts the data using its private key. Because the private key is retained at the first station, and is not practically discoverable

by other parties, PPK schemes are considered to be secure. However, the encryption and decryption processes are relatively slow, so that such schemes are not practical for encryption of real-time high-speed data, such as television program signals, for which distribution networks are primarily intended.

5 In symmetric key schemes, a single key, referred to as a working key, is used by both of first and second stations to encrypt and decrypt data being communicated between the stations. The nature of the working key is such that encryption of real-time high-speed data, such as television program signals, is practical. However, these schemes require that the working key be present in both stations, and make it desirable for the
10 working key to be periodically changed or updated. Thus symmetric key schemes require generation of a working key in one of the stations or in a third station referred to as a key distribution agent, and communication of the working key to the other station(s).

This communication itself presents a risk of the working key being insecure, and this risk increases with the frequency with which the working key is updated. It is also
15 known to avoid this risk by using a PPK scheme for communication of a working key, and then to use the working key for data encryption.

An object of this invention is to provide a method of facilitating secure communications in a distribution network.

Summary of the Invention

20 One aspect of this invention provides a method of facilitating secure communications using encryption and decryption processes in a distribution network comprising a central station and a plurality of addressable end stations, in which communications from the central station addressed to and intended for a particular end station are delivered via the network to a plurality of end stations, wherein the central
25 station has, and one or more of the end stations can each have, a respective public and private key (PPK) of a PPK encryption scheme, comprising the steps of:

(a) determining in communications between the central station and an end station whether the end station has a PPK, if so proceeding with step (b) and if not proceeding with step (c);

30 (b) at the central station, determining the public key (PK) of the end station, generating a working key (WK) for encryption of communications to the end station, encrypting the WK using the PK of the end station, and communicating the encrypted WK to the end station; at the end station, decrypting the WK using the private key of the end station; and proceeding with step (d);

35 (c) at the end station, determining the public key (PK) of the central station, generating a working key (WK) for encryption of communications to the central station, encrypting the WK using the PK of the central station, and communicating the encrypted WK to the central station; at the central station, decrypting the WK using the private key of the central

station; and proceeding with step (d);

(d) using the WK to encrypt at the central station, and to decrypt at the end station, communications from the central station to the end station.

Another aspect of this invention provides a method of facilitating secure communications in a distribution network comprising a central station and a plurality of addressable end stations, in which communications from the central station addressed to and intended for a particular end station are delivered via the network to a plurality of end stations, wherein the central station has a public and private key (PPK) of a PPK encryption scheme and each end station has an individual identity (ID) and an individual cryptographic signature encrypted using a private key of a predetermined PPK encryption scheme, comprising the steps of: communicating the ID of an end station to the central station; at the end station, generating a working key (WK) for encryption of communications between the end station and the central station and encrypting the WK using the public key of the central station; communicating the encrypted WK from the end station to the central station; at the central station, decrypting the encrypted WK using the private key of the central station; communicating the cryptographic signature of the end station to the central station; and at the central station, decrypting the cryptographic signature using a public key of the predetermined PPK scheme for authentication of the end station.

20 Brief Description of the Drawings

The invention will be further understood from the following description with reference to the accompanying drawings, in which:

Fig. 1 illustrates parts of a distribution network to which the invention is applied; and

25 Fig. 2 is a flow chart illustrating steps of a method for facilitating secure communications in the network in accordance with the invention.

Detailed Description

The invention is described below in the context of a hybrid fiber-coax (HFC) distribution network in which signals are distributed from a central station or head end (HE) to a large number of subscriber end stations (ES) via optical fibers and coaxial cables in known manner. An example of such a network is described in Warwick United States Patent No. 5,408,259 issued April 18, 1995 and entitled "Data Modulation Arrangement For Selectively Distributing Data". Typically in such a network digital data communications are provided between any ES and the HE using asynchronous transfer mode (ATM) cells which are communicated in both directions, i.e. downstream from the HE to the ES and upstream from the ES to the HE, using suitable modulation schemes and carrier frequencies outside the bands used for analog television signals also carried on

the coaxial cables. However, it is observed that the invention is equally applicable to other forms of distribution network.

Referring to Fig. 1, there is illustrated parts of a distribution network in which many end stations, only two of which are shown and are referenced 10 and 12, are connected via branched cables 14 of the distribution network to a head end 16, via which the end stations have access to a network (not shown) which for example supplies digital television program signals subscribed to by end station subscribers. The cables 14 can comprise both optical fiber and coaxial cables forming a hybrid fiber-coax arrangement, on which the digital signals can be communicated in known manner using ATM cells.

As can be appreciated from the illustration in Fig. 1, signals communicated by the head end 16 and intended for any particular end station will actually be delivered via the cables 14 to all of the end stations. For secure and/or private communication of the signals, the head end 16 includes an encryption engine 18 which encrypts the signals in accordance with a working key known only by the head end and the intended end station, which also includes an encryption engine 20 which decrypts the signals for use. These working keys are similarly used for communications in the opposite direction, from the end station to the head end 14. The working keys of this symmetric key encryption scheme are provided in the head end and the end station in a manner which is described in detail below.

The end stations 10 and 12 are of two types, with differing levels of security to enable different security needs of subscribers to be accommodated. The end station 12 represents a relatively secure end station, which includes its own public and private keys of a PPK encryption scheme. As explained in the introduction, such an end station has a relatively high complexity and cost, because of the need for secure storage of the keys and operation of the PPK encryption. Other end stations, which do not have their own public and private keys and accordingly can be provided at a much lower cost, are represented by the end station 10. The network as a whole may have an arbitrary mix of these two types of end station.

Each end station 10 or 12 also has an individual, unique identity number, which is stored (e.g. hard wired) into the ES during its manufacture. This is referred to as a global ID (identity). The global IDs of all of the end stations are stored in a database 22, which can be colocated with the head end 16 or separately from it and with which the head end 16 communicates via a path 24. The head end 16 also has its own public and private keys of a PPK encryption scheme.

Fig. 2 shows steps of a process which is followed in order to set up secure communications between the head end 16 and one of the end stations 10 or 12. This process takes place between the head end and the respective end station without involvement of any other node such as a central key distribution agent, and is described

below as being initiated in each case following any reset (e.g. following a power-up) of either the head end 16 or the respective end station. Consequently, the working key which is used for encrypting the communications between the head end and the end station is changed on any reset. However, the same process can alternatively or
5 additionally be carried out on demand, and/or periodically to provide periodic changes of the working key. It is also observed that the encrypted communications take place between the encryption engines 18 in the head end 16 and 20 in the respective end station 10 or 12, and communications on the network access side of the head end 16 are not subject to the same encryption.

10 In Fig. 2, a block 30 represents a reset of the head end (HE) or end station (ES), in response to which, as shown by a block 31 in Fig. 2, a dialog or handshake is carried out between the HE and the ES to establish communications between them. These communications are effected using unencrypted ATM cells using addresses of the end station and the head end. As a part of this dialog, as shown by a block 32 in Fig. 2 the
15 head end 16 interrogates the end station to determine whether or not the end station has its own public and private keys. If not, i.e. if the end station is an end station 10 as described above, then the process continues with successive blocks 33 to 38 in Fig. 2. If the interrogation establishes that the end station is an end station 12 having its own public and private keys, then the process instead continues with blocks 39 to 44 in Fig. 2.

20 In the former case of an end station 10, as shown by the block 33 the head end 16 communicates its public key (PK) to the end station 10; this communication can form part of the dialog block 31. The end station 10 randomly generates (block 34) a working key (WK) for communicating signals in a symmetric key encryption scheme, and encrypts (block 35) this working key in accordance with the supplied public key, sending the
25 encrypted working key in a message to the head end 16. The head end 16 decrypts (block 36) the encrypted working key from this message in accordance with its private key, which is not known to others so that the communication of the working key from the end station 10 to the head end 16 is secure, and optionally but preferably sends an acknowledgement to the end station 10. As shown by the block 37, the head end 16 and
30 the end station 10 then load their encryption engines 18 and 20 respectively with the working key, and thereafter (until this process is repeated, for example in response to a subsequent reset at either end) communications between them take place with data encrypted in accordance with the working key. An optional additional step represented by the block 38 provides for authentication of the end station 10 in a manner described
35 below.

Conversely, in the latter case of an end station 12, as shown by the block 39 the end station 12 communicates its public key (PK) to the head end 16; this communication can form part of the dialog block 31. An optional authentication step for the end station

12 can be carried out by the head end 16 as represented by the block 40 in a manner described below. The head end 16 randomly generates (block 41) a working key (WK) for communicating signals in a symmetric key encryption scheme, and encrypts (block 42) this working key in accordance with the supplied public key of the end station 12, sending the encrypted working key in a message to the end station 12. The end station 12 decrypts (block 43) the encrypted working key from this message in accordance with its private key, which is not known to others so that the communication of the working key from the head end 16 to the end station 12 is secure, and optionally but preferably sends an acknowledgement to the head end 18. As shown by the block 44, the head end 16 and the end station 12 then load their encryption engines 18 and 20 respectively with the working key, and thereafter (until this process is repeated, for example in response to a subsequent reset at either end) communications between them take place with data encrypted in accordance with the working key.

It can be seen from the above description that, in the relatively secure but more expensive situation in which the end station 12 includes its own public and private keys, these are used for communicating a working key generated in the head end, whereas in the other case the end station 10 generates the working key and this is communicated to the head end using the latter's public key.

The optional step of authentication of the end station 12 in the block 40 as described above can make use of the global ID of the end station 12 together with data in the database 22, in which the public key of the end station 12 is stored in association with this global ID. As part of the dialog block 31, the end station communicates its global ID to the head end 16. In the step 40, therefore, the head end 16 can communicate via the path 24 with the database 22 to confirm that the public key which it has received from the end station 12 in the step 39 matches that stored in the database 22 for this end station's global ID, the subsequent steps 41 to 44 only being followed if this authentication step is successful.

Alternatively, or in addition, the optional end station authentication step of block 40 can comprise the steps of the head end sending an unencrypted message to the end station 12 with a request that it be cryptographically signed. In accordance with this request, the end station 12 produces a digest of the message using a known hashing function (thereby reducing the data to be encrypted), encrypts this digest in accordance with its private key, and sends the encrypted message digest to the head end 16. The head end 16 then decrypts this in accordance with the public key of the end station, retrieved from the database 22, to confirm the digest of its original message which the head end also produces using the hashing function.

It can be seen that, alternatively, the steps represented by the blocks 39 and 40 in Fig. 2 could be replaced by a single step in which the head end 16 determines the public

key of the end station 12 from the database 22 in accordance with the global ID of the end station 12 supplied in the dialog 31, without any authentication of the end station or any communication of the public key from the end station 12.

5 The above sequences provide a particularly strong or secure authentication of the end station 12. For the end station 10 which does not have its own public and private keys, a weaker but still valuable authentication can be provided as shown by the block 38. The authentication block 38 is shown in Fig. 2 as the final block in the process because this enables the exchange of data in the authentication process to be encrypted in accordance with the working key, but this authentication step could alternatively be
10 provided anywhere else in the sequence of steps from the blocks 31 to 37.

For this optional authentication step, the end station 10 is manufactured (e.g. hard wired) with not only its global ID, but also a cryptographic signature. Conveniently, the end station 10 is manufactured with a certificate comprising data including the global ID of the end station and the public key of the manufacturer and a cryptographic signature
15 comprising an encryption, in accordance with the private key of the manufacturer, of a digest of that data produced using a known hashing function. The public key of the manufacturer can also or instead be stored in the database 22. The optional end station authentication step of the block 38 comprises a communication of the cryptographic signature from the end station 10 to the head end 16 (as explained above this could be a
20 part of the dialog 31 or any later step, but the encryption after the block 37 obstructs public observation in the network of cryptographic signatures). The head end 16 then confirms the authenticity of the end station 10 by decrypting the cryptographic signature using the manufacturer's public key, producing a digest from the same data (global ID and public key, both of which can be communicated in the dialog step 31 or later) and the
25 known hashing function, and matching these.

This is a relatively weak authentication, in that identical copies of the end station 10, including duplicated data and cryptographic signatures, could operate at different times on the network without this being detected. However, simultaneous operation of two or more such duplicates would be detected by the fact that two or more end stations
30 would be supplying the same global ID which is supposedly unique. Thus even such a weak authentication is valuable especially in detecting illicit large-scale duplication of end stations.

The processes in accordance with the invention as described above provide a number of significant advantages over known configurations. In particular, requirements
35 for secure storage of public and private keys are minimized in the network as a whole, and eliminated for the end stations 10 which can accordingly be provided at relatively lower cost. At the same time, end stations 12 with greater security can be provided, and the head end 16 can operate simultaneously with both types of end station. This, combined

with optional authentication of the end stations as described above, enables different degrees of security to be easily provided in the network in accordance with service requirements.

5 Furthermore, renewal of the working keys at reset is simpler than providing time-based schedules for changing encryption keys, and key exchanges take place only between the head end and the end station which use the keys, thereby enhancing security compared with distribution of keys from a key distribution agent. In addition, all of the data flowing between the head end and any particular end station 10 or 12, between successive resets, can be encrypted using a single working key, thereby simplifying the
10 encryption and decryption processes. However, it is observed that different working keys could be generated, communicated, and used in the same manner as described above for encrypting and decrypting different types of information, or different services, for a single end station 10 or 12.

15 Although particular embodiments of the invention have been described in detail, it should be appreciated that numerous modifications, variations, and adaptations may be made without departing from the scope of the invention as defined in the claims.

WHAT IS CLAIMED IS:

1. A method of facilitating secure communications using encryption and decryption processes in a distribution network comprising a central station and a plurality of addressable end stations, in which communications from the central station addressed to and intended for a particular end station are delivered via the network to a plurality of end stations, wherein the central station has, and one or more of the end stations can each have, a respective public and private key (PPK) of a PPK encryption scheme, comprising the steps of:
- 5 (a) determining in communications between the central station and an end station whether the end station has a PPK, if so proceeding with step (b) and if not proceeding with step (c);
- 10 (b) at the central station, determining the public key (PK) of the end station, generating a working key (WK) for encryption of communications to the end station, encrypting the WK using the PK of the end station, and communicating the encrypted WK to the end station; at the end station, decrypting the WK using the private key of the end station; and proceeding with step (d);
- 15 (c) at the end station, determining the public key (PK) of the central station, generating a working key (WK) for encryption of communications to the central station, encrypting the WK using the PK of the central station, and communicating the encrypted WK to the central station; at the central station, decrypting the WK using the private key of the central station; and proceeding with step (d);
- 20 (d) using the WK to encrypt at the central station, and to decrypt at the end station, communications from the central station to the end station.
2. A method as claimed in claim 1 wherein each end station has an individual identity (ID) and step (a) includes the step of communicating the ID of the end station to the central station.
- 25 3. A method as claimed in claim 2 wherein in step (b) the PK of the end station is determined by the central station from a database using the ID of the end station.
4. A method as claimed in claim 1, 2, or 3 wherein step (b) further comprises an end station authentication step comprising the steps of communicating an unencrypted message from the central station to the end station, producing an encrypted message at the end station using the private key of the end station, communicating the encrypted message to the central station, decrypting the message at the central station using the PK of the end station, and comparing the decrypted message with the original message.
- 30 5. A method as claimed in claim 4 wherein in step (b) the end station authentication step is carried out before the step of communicating the encrypted WK to the end station.
- 35

6. A method as claimed in any of claims 1 to 5 wherein in step (b) the PK of the end station is communicated to the central station from the end station.
7. A method as claimed in claims 2 and 6 wherein in step (b) the PK of the end station is verified by the central station from a database using the ID of the end station.
- 5 8. A method as claimed in any of claims 1 to 7 wherein a plurality of end stations which do not have a PPK each have an individual cryptographic signature encrypted using a private key of a predetermined PPK scheme, step (a) or (c) includes the step of communicating the cryptographic signature of the end station to the central station, and step (c) further comprises an end station authentication step comprising, at the central station, decrypting the cryptographic signature using a public key of the predetermined PPK scheme.
- 10 9. A method as claimed in claims 2 and 8 wherein the individual cryptographic signature comprises an encryption of data derived from the ID of the respective end station.
- 15 10. A method as claimed in claim 8 or 9 wherein the predetermined PPK scheme uses a private key and a public key of a source of the end station.
11. A method as claimed in claim 8, 9, or 10 wherein the cryptographic signature is communicated to the central station in step (c).
- 20 12. A method as claimed in claim 11 and including the steps of encrypting the cryptographic signature at the end station, and decrypting the encrypted cryptographic signature at the central station, using the WK.
13. A method as claimed in any of claims 1 to 12 and further comprising the step of using the WK to encrypt at the end station, and to decrypt at the central station, communications from the end station to the central station.
- 25 14. A method of facilitating secure communications in a distribution network comprising a central station and a plurality of addressable end stations, in which communications from the central station addressed to and intended for a particular end station are delivered via the network to a plurality of end stations, wherein the central station has a public and private key (PPK) of a PPK encryption scheme and each end station has an individual identity (ID) and an individual cryptographic signature encrypted using a private key of a predetermined PPK encryption scheme, comprising the steps of:
- 30 communicating the ID of an end station to the central station;
at the end station, generating a working key (WK) for encryption of communications between the end station and the central station and encrypting the WK

using the public key of the central station;

communicating the encrypted WK from the end station to the central station;

at the central station, decrypting the encrypted WK using the private key of the

central station;

5 communicating the cryptographic signature of the end station to the central station;

and

at the central station, decrypting the cryptographic signature using a public key of
the predetermined PPK scheme for authentication of the end station.

15 15. A method as claimed in claim 14 wherein the individual cryptographic signature
comprises an encryption of data derived from the ID of the respective end station.

16. A method as claimed in claim 14 or 15 wherein the predetermined PPK scheme
uses a private key and a public key of a source of the end station.

17. A method as claimed in claim 14, 15, or 16 wherein the step of communicating the
cryptographic signature of the end station to the central station comprises the steps of
15 encrypting the cryptographic signature at the end station using the WK, communicating
the encrypted cryptographic signature from the end station to the central station, and
decrypting the encrypted cryptographic signature at the central station using the WK.

18. A method of facilitating secure communications in a distribution network,
substantially as hereinbefore described with reference to Figs 1 and 2 of the
20 accompanying drawings.



The Patent Office

12

Application No: GB 9700921.1
Claims searched: 1-13

Examiner: Mr B J Spear
Date of search: 19 March 1997

Patents Act 1977 Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:
UK Cl (Ed.O): H4P (PDCSC)
Int Cl (Ed.6): H04L 9/30
Other: Online: WPI, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
	NONE	

X	Document indicating lack of novelty or inventive step	A Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E Patent document published on or after, but with priority date earlier than, the filing date of this application.



The
Patent
Office

13

Application No: GB 9700921.1
Claims searched: 14-17

Examiner: Mr B J Spear
Date of search: 21 May 1997

Patents Act 1977
Further Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:
UK CI (Ed.O): H4P (PDCSA)
Int CI (Ed.6): H04L 9/32
Other: Online: WPI, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	EP0328232A2 (Fischer)	-
A	WO 95/23468A1 (Merdan)	-

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

(12) UK Patent Application (19) GB (11) 2 316 503 (13) A

(43) Date of A Publication 25.02.1998

(21) Application No 9617596.3

(22) Date of Filing 22.08.1996

(71) Applicant(s)
ICL Personal Systems Oy

(Incorporated in Finland)

PO Box 458, SF-00101 Helsinki, Finland

(72) Inventor(s)
Tapani Lindgren

(74) Agent and/or Address for Service
S M Dupuy
International Computers Limited, Cavendish Road,
STEVENAGE, Hertfordshire, SG1 2DY,
United Kingdom

(51) INT CL⁶
G06F 1/00

(52) UK CL (Edition P)
G4A AAP

(58) Documents Cited
GB 2236604 A EP 0332304 A2 WO 93/11480 A1
US 5375206 A US 4924378 A

(58) Field of Search
UK CL (Edition O) G4A AAP
INT CL⁶ G06F

(54) Software licence management

(57) A software licence management method and system is for a computer system including at least one server (1,5) and particularly for a plurality of computers connected via a network. Before a service (2) can offer functionality to a user it has to check that the user has a licence for that service. A licensing subsystem (3) is associated with it a ticket database (4) that hold tickets corresponding to existing licences. Tickets, if available, are issued to a service on request, thereby verifying the existence of a licence. The receipt of a ticket allows a service to offer functionality.

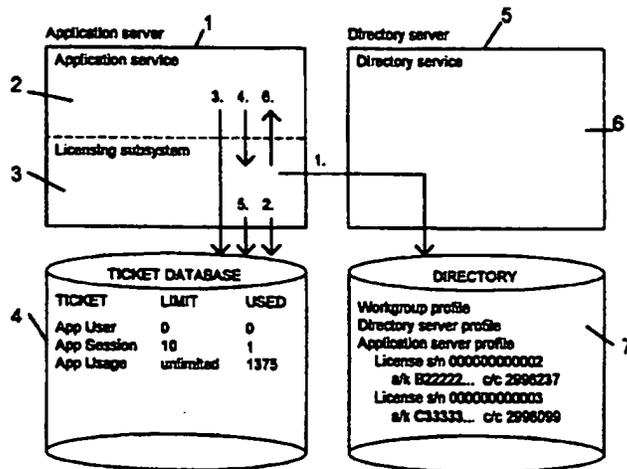


FIG 1

GB 2 316 503 A

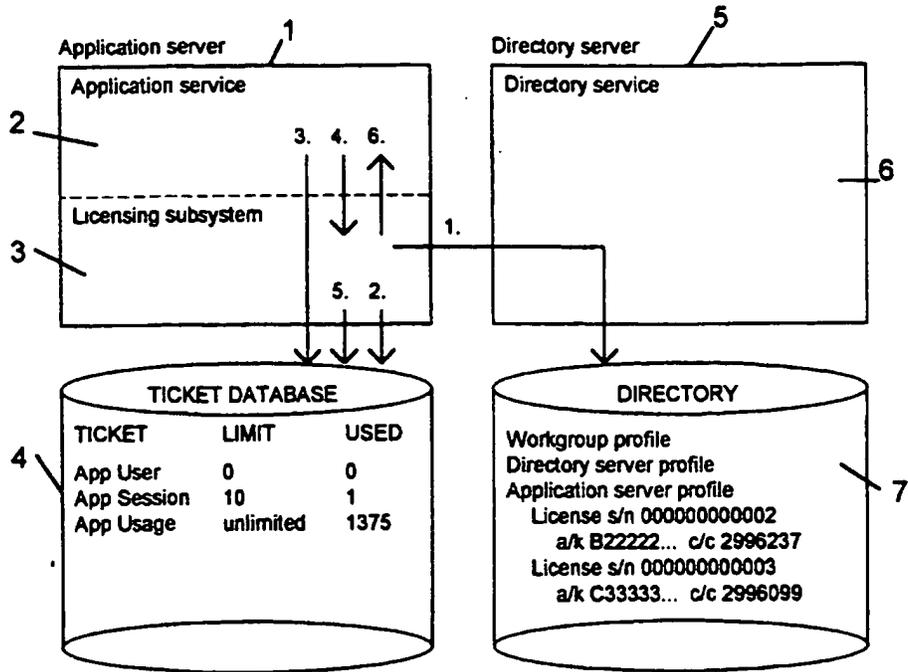


FIG 1

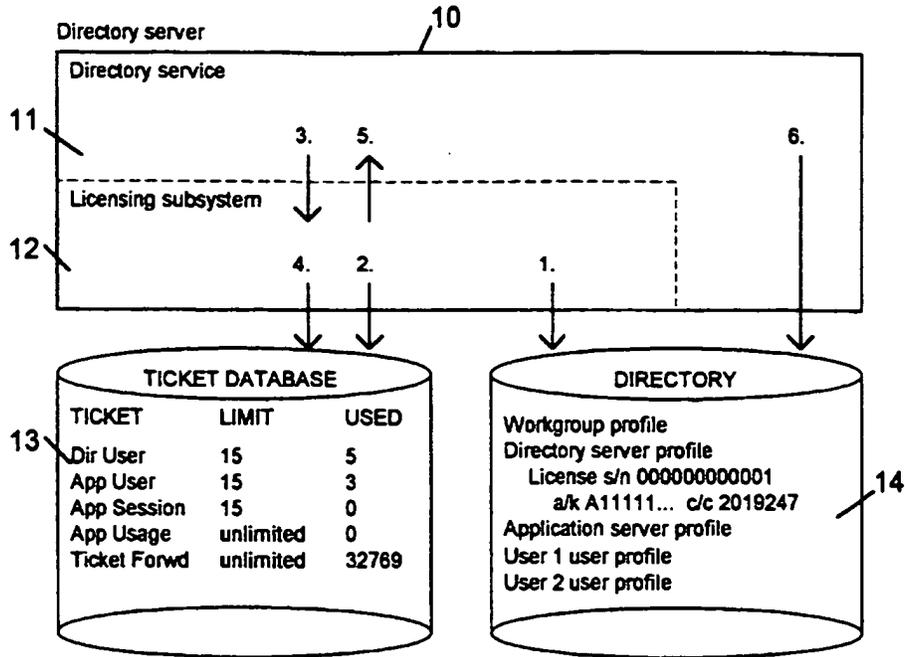


FIG 2

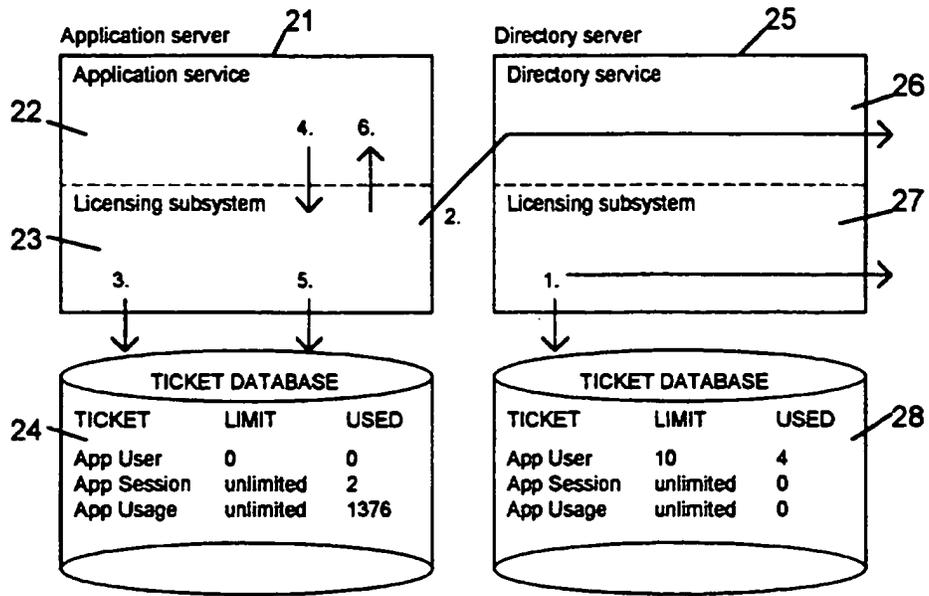


FIG 3

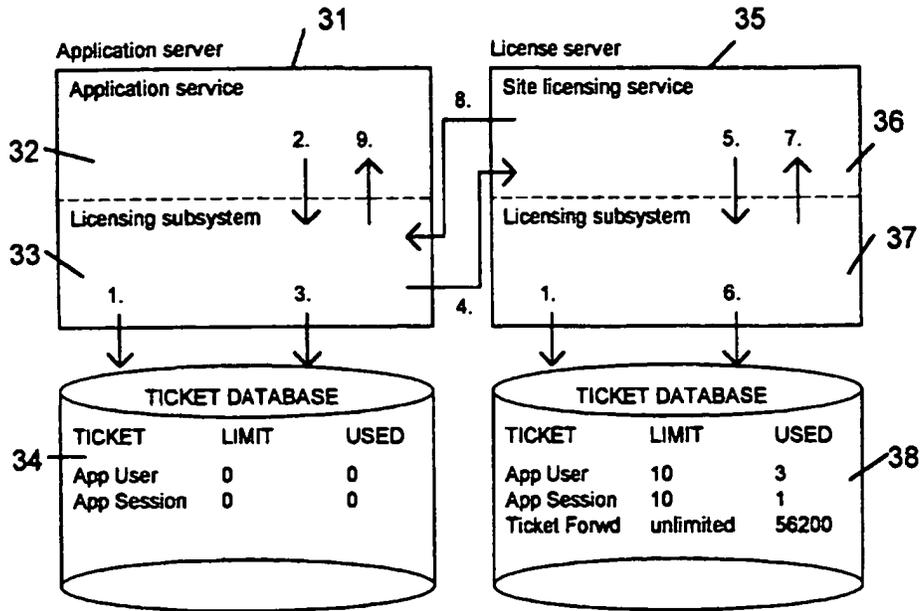


FIG 4

2316503

SOFTWARE LICENCE MANAGEMENT

This invention relates to software licence management and in particular to licence management for software running on a plurality of computers connected via a network.

Conventionally, licences have been provided by software vendors as separate licences for individual workstations or as a single licence for a number of workstations. Various schemes have been proposed in order to try and make unlicensed software unusable, in particular pirated (illegal) copies of software. Other schemes have been proposed such as in order to achieve low initial software costs but licensing royalties consistent with the extent of use, in order not to deter low-usage users from purchasing particular forms of software, and thus to reduce piracy, whilst still enabling a vendor to collect higher dues from high-usage users.

The present invention is, particularly, concerned with a distributed system consisting of various server and client programs running on various computers which are connected via a local or wide area network, and an object is to provide server software licensing which ensures that all software running in the network has been purchased legally.

According to one aspect of the present invention there is provided a software licence management method for use with a computer system including at least one server, the method being such that before a service can offer functionality to a user, the said service shall verify that the user has a licence for said service, and wherein the computer system further includes a licensing subsystem with which are associated service tickets corresponding to existing licences, the method including the steps of the said service requesting a respective service ticket from the licensing

subsystem prior to offering functionality to the user, and the licensing subsystem issuing a said service ticket to the said service, if one is available, thereby verifying the licence exists and allowing the said service to offer functionality.

According to another aspect of the present invention there is provided a computer system including at least one server and a software licence management system, the management system being such that before a service can offer functionality to a user, the service shall verify that the user has a licence for said service, the management system including a licencing subsystem with which are associated service tickets corresponding to existing licences, and the management system being such that a said service ticket is issued to a service, if one is available, upon request by the service, thereby verifying existence of a licence and allowing the said service to offer functionality.

Embodiments of the invention will now be described with reference to the accompanying drawings, in which:

Figure 1 Illustrates obtaining a session or usage ticket for an application,

Figure 2 Illustrates obtaining a user ticket for a directory server,

Figure 3 Illustrates independent licence sharing, and

Figure 4 Illustrates licence sharing with a site licencing service,

Various terms used in the following will first be defined. For the purposes of the description the software is considered to relate to a Groupware Office system which provides various facilities including mail, for example.

Definitions

- "Server" An instance of server software running on a server computer. Usually only one such instance runs on any one computer. Each "server" implements one or more collections of related functions called "function sets", examples of which are directory, mail, library etc. The "directory function set" includes functions to access a database that contains information about the Groupware Office system.
- "Client" Any piece of software that connects to the "server" using a "client-server protocol" to access the functions offered by the "server". A "client" may be a program run by a user on a workstation, or a part of any other program.
- "Session" An instance of client-server dialogue between one "client" and one "server". Each "session" allows the "client" to use the functions of one or more "function sets".
- "Directory Server" A server that implements the directory function set.
- "Mail Server" A server that implements the mail function set. [A server may be a directory server and a mail server simultaneously.]
- "User" A person (actual or virtual) listed in the database of a directory server.

- "User profile" The information pertaining to one user stored in a directory database entry, such as the name of the user, user authentication information, the list of servers and function sets the user is permitted to access, etc.
- "Server profile" The name and network address of a server and the list of services offered by it, as stored in a directory database entry.
- "Service profile" Information stored in a directory database entry about one service in one server. If the same type of service is offered by more than one server, each instance has its own profile.
- "Site profile" Information stored in a directory database entry about one site.
- "Site" A set of servers connected to a single directory server. Each server belongs to exactly one site, and each site has exactly one directory server. Other servers in the site are optional, usually unlimited in number, and sometimes called member servers or application servers.
- "Enterprise" A set of sites that share their directory databases. The directory servers in each site replicate directory information to other directory servers in the enterprise. Each directory server contains both "local" and "external" information. One of the directory servers, the "enterprise directory server" controls the others, which are

"site directory servers".

"Subsystems"

Collections of programs and/or subroutines that perform a set of interrelated functions. Some subsystems implement a function set within a server program, while others run independently as stand-alone applications. Many subsystems are collections of common subroutines called by other subsystems. Client programs are also subsystems.

"Subsystem id"

A respective unique number identifying every type of subsystem. Some systems use two ids, a "real subsystem id" when dealing with licensing issues and an "alias subsystem id" when performing a task on behalf of a virtual entity, such as "generic gateway no 9".

Not every subsystem software needs to be purchased individually. Most collections of subroutines can be used freely by other subsystems.

"Services"

Those subsystems that need to be explicitly purchased.

Service types may include directory service, mail service, fax gateway, X.400 gateway, enterprise option, library service, power library option, etc. Each service is located in one server, either as a function set of the server program or a standalone application running in the same computer. Many services of the same type can exist in different servers.

The software licence management system of the present invention proceeds from the premise that before offering any usable functionality to the users, services shall verify from a "licensing subsystem" that a licence for the service exists. To achieve that, it is proposed that the licensing subsystem holds "service tickets" and a service requests a permission to offer its functions to the user by requesting a corresponding "service ticket" be provided from the licensing subsystem. Each service knows that kind of tickets are needed to fulfil the service's functionality. The licensing subsystem has to keep track of the available licences and of the service tickets it has issued. A service ticket may be considered as partially the equivalent of a password in that one must be provided before a service can operate.

A "licence" is a permission to use one or more services within certain limits. Typically these limits are "license duration", which specifies the maximum length of the period when the licence may be used (the "active" period) and the "licence size", which specifies the maximum number of users of the licence. The interpretation of "number of users" varies from service to service. It may, for example, mean the number of users in the local directory that are allowed to use the service, or the number of concurrent sessions that are connected to the service. Licence duration and/or size may also be unlimited.

When a customer purchases a Groupware, for example, software product which employs the software licence management method and system of the invention from a supplier, as well as the media containing the software itself and associated documentation, there is obtained a single licence to one or more services. Each said product has a unique serial number. The license is supplied in the form of a licence agreement document on which the licence information is printed. This licence information consists of the serial number of the product and an "activation key" for the licence. The licence

size and duration and the included services are encoded into the activation key.

The software license management method and system of the invention is such that the Groupware software may be copied and installed by the customer without any technical restrictions, but before any of the services can be used, a corresponding licence must be installed and activated. Licence installation consists of entering the license information (serial number and activation key) into the server profile of a server in the directory server's database, ie in the site directory, in the server profile of the server in question. Licence activation consists of setting the active period of the installed licence so that service tickets can be issued. Typically licence installation and licence activation are performed simultaneously by the server setup program. The license information is stored in the site directory, in the server profile of the server in question.

As will be appreciated, there also exists "evaluation licences" which allow a prospective customer to use a service for a short trial period before actually purchasing it. These licences typically have a very short duration and a relatively small size. The product serial numbers associated with such evaluation licences are not necessarily unique, since the licence information may be distributed on CD-ROMs or via public networks.

As mentioned above, each software product contains just one licence, although that licence may include a large number of services, for example, enough to build a complete Groupware Office site with all of the basic services. Alternatively, the licence may include just one service. Product with that kind of licence could be used to expand the capacity or functionality of an existing Groupware Office System.

The core of the process of designing a product is, therefore, determining what services will be included and the size and duration of the licence. This information is encoded into a number, the covert code, which may be a 7-digit number, for example. The building of the covert code is discussed in more detail hereinafter.

The amount of information that can be encoded into the covert code is limited by the size of the code. Therefore, there are some necessary restrictions on what kinds of licences are possible. The most obvious limitation is that the size and duration can only take certain discrete values. Also, the same size and duration will apply to all services covered by the licence. Another restriction is that only the most common groups of services can be combined freely into a multi-service licence. Other services will have to be licensed individually.

The covert code, which specifies the properties of the software licence, is thus a part of the product description in the logistics database. When the product is manufactured it has the unique serial number, referred to above, assigned to it. The actuation key for the license is calculated as a function of the serial number and the covert code using a secret algorithm. The serial number and activation key may be printed on a label, which is attached to the licence agreement document.

When creating a site, a customer must have a licence that includes a site creation ticket. This licence is installed for the directory server. The customer may also install additional licences for the directory server and for other servers. Each licence may apply to one or more services. Some licences are valid only in that server for which they are installed, whilst other licences may be shared with other servers at the same site (see later). Shared licences would usually be installed in the server profile of the directory

server, although optionally, and with some restrictions, another server may be designated as the licence server. The serial number of the first licence installed in a directory server's profile can be used to identify the site uniquely. Thus the directory server is computer number 1. Other servers in the site will use the same site id but differing computer numbers for identification.

When a service program is about to execute an action which requires that a customer possesses a licence for that service, the service program must first obtain a corresponding service ticket from the licensing subsystem. The actions concerned are ones which are potentially profitable for the customer and may, for example, include namely:

setting up a new Groupware site;

creating a new user account;

setting up an instance of the mail service;

enabling mail usage for a user and creating a user mailbox;

starting a session between a mail UI client and the mail server;

sending a mail message;

relaying a mail message to an X.400 mail network.

Each kind of action requires a specific kind of service ticket. To obtain a ticket the service needs to specify the ticket type and the number of tickets. The service tickets are only identified by ticket type. There is a licensing subsystem in each server and it counts the number of

different tickets in all available licences and keeps track of how many licences of each type are being used in the server.

The steps involved in obtaining various licences will now be described in greater detail. With respect to Figure 1 there will, firstly, be described the case of an application service running in a separate server from the directory server obtaining a session or usage ticket.

Figure 1 illustrates schematically an application server 1, providing an application service 2 and having a licensing subsystem 3, with an associated ticket database 4, a directory server 5 providing a directory service 6 and having an associated directory 7. The ticket database 4 has stored therein details of ticket types, the limit, if any, of the number of such tickets which are available and the number of used tickets for each type. The ticket types as illustrated are "App User" (Application User), "App Session", "App Usage". The directory 7 has stored therein, the "Workgroup profile", the "Directory server profile", the Application server profile. In the example illustrated, the application server has two associated licenses whose serial numbers (s/n) are 00000000002 and 00000000003, respectively, whose activation keys (a/n) are of the form B2222... and C33333..., respectively, and whose covert codes (c/c) are 2996237 and 2996099, for example, respectively.

When the licensing subsystem 3 on the application server starts, it fetches the application server's server profile from the directory service 6, 7 using the directory API (Application Programming Interface) (Step 1 in Figure 1). The licensing subsystem 3 analyses the licences and updates the limits of each ticket type in the local ticket database 4. The numbers of used tickets are not modified at this time, the old accumulated values being maintained (step 2).

When the application service 2 starts, and before any user logs in, it tells the licensing subsystem 3 to set the number of used session tickets to zero. This frees any session tickets that may have been left unreturned at the end of a session because of a system crash etc. (Step 3). The application service 2 then requests a service ticket (session or usage) from the licensing subsystem 3, since without a ticket it cannot proceed. (Step 4). The licensing subsystem checks the ticket availability in the local ticket database 4 and updates the used ticket count (step 5), to take into account the requested ticket, before issuing the ticket to the application service (step 6), which then proceeds since it has determined that there exists the appropriate licence.

In the embodiment of Figure 2, the procedure whereby a directory service obtains a ticket for adding a user to a directory is illustrated.

A directory server 10 provides a directory service 11 and includes a licensing subsystem 12 with an associated ticket database 13, the directory service 11 having an associated directory 14. The ticket database 13 has stored therein details of ticket types, the limit, if any, of the number of such tickets which are available, and the number of used tickets for each type. The ticket types are illustrated as "Dir User" (Directory User), "App User", "App Session", "App Usage" and "Ticket Forwd" (Ticket Forwarding). The directory 14 has stored therein the "Workgroup profile", the "Directory server profile", the "Application Server profile" and the user profile of two users, User 1 and User 2. The Directory server has a license serial number (s/n) 000000000001, with an actuation key (a/) of the form A11111..., and a corresponding covert code (c/c) 2019247, for example.

When the licensing subsystem 12 on the directory server 10 starts, it fetches the directory server's server profile directly from the directory 14 (step 1). The licensing

subsystem 12 analyses the licenses and updates the limits of each ticket type in the ticket database 13. The numbers of used tickets are not modified at this time, the old accumulated values being maintained (step 2).

When it is desired to add a new user to the directory, the directory service 11 requests a user ticket from the licensing subsystem 12 (step 3). The licensing subsystem 12 checks ticket availability in the local ticket database 13 and updates the used ticket count (step 4) to take into account the requested ticket. The licensing subsystem 12 issues the requested user ticket to the directory service 11 (step 5). The directory service then adds the new user to the directory 14, that is it adds its user profile.

To ensure consistency, the directory service 11 may periodically count the number of users in the directory 14 and tell the licensing subsystem 12 to set the used ticket count accordingly.

When a licence is installed, the start time of its active period will be fixed. By default this is the same as the installation time, but any time in the past or in the future may be specified. If the licence has a limited period, the end time will also be set. The licence will be active whenever the current time is after the start time and before the end time.

A customer may wish to deactivate a licence so that it cannot be used. Thus can be done at any time by altering the end time of the licence with the server setup program. The end time may be altered freely, as long as the active period does not exceed the licence duration.

Once installed, limited-duration licences are fixed, ie they cannot be removed, except by remaining the entire site directory, or moved to another server, and their start time

may not be altered. These restrictions, however, do not apply to unlimited-duration licences. They may be removed, reinstalled, moved or altered freely. The only restriction that remains is that a licence may only be installed for one server at a time.

A further restriction applies to the licence that has been used to create a site. This licence cannot be removed or deactivated, except by removing the entire site.

The licensing method described with reference to Figures 1 and 2 applies only to local licences, ie the tickets included in a license can only be issued in one server, the server whose server profile contains the licence. Often there is a need to share a single licence between two or more servers, so that tickets can be issued in all of them. Most commonly, the user tickets for an application are needed in the directory server, and session and usage tickets in the application servers.

If a licence includes an unlimited number of a certain kind of service ticket, sharing the licence is not very complicated. Any server can read the licences in any other server's profile. If the licensing subsystem in a server can verify that another server's licence contains an unlimited supply of freely shareable tickets, it will deduce that these tickets may be issued without limit in any server, independently of other servers. This is independent license sharing.

Not all licences are necessarily shareable, even if they contain an unlimited number of tickets. Whether each licence is shareable or not is a licence-specific property, which is coded in the covert code together with other licence properties.

The first implementation of the licensing subsystem capable

of independent licence sharing will not scan every server profile for available licences. It will only scan its own server profile and the directory server's profile. Therefore, all licenses that are meant to be shared, should be installed for the directory server.

If the tickets to be shared are limited in number, the situation is more complicated. For each "pool" of shareable tickets, there must be a single process that is responsible for keeping track of their usage. It has to co-ordinate the activities of the licensing subsystems in various servers and make sure that no ticket is issued more than once. To achieve this a site licensing service can be implemented. This is an extension to the licensing subsystem that allows the licensing subsystems of various servers to communicate using a client-server protocol. The site licensing service, together with the licensing subsystem in the same server, control the usage of tickets installed for that server. Another server's licensing subsystem may connect to the site licensing service and ask the latter to obtain a service ticket on its behalf.

Licenses that are shareable by independent sharing would also be shareable by the site licensing service, with the addition that also limited-number tickets could be shared. Some types of licences will still be unshareable, since shareability is a licence-specific property. The licensing service could itself require a licence. A site licensing service could be expanded to support also client licensing and enterprise-wide licence sharing.

An example of independent licence sharing will now be described with reference to Figure 3 in which an application server 21 provides an application service 22 and includes a licensing subsystem 23 with an associated ticket database 24. A directory server 25 provides a directory service 26 and includes a licensing subsystem 27, with a associated ticket

database 28, and a directory (not shown but containing information of the type illustrated in Figures 1 and 2). The ticket databases 24 and 28 have details of ticket type, limit and usage as indicated.

The licensing system 27 on the directory server 25 fetches the server profile from the directory (not shown), analyses the licences therein, and updates the ticket limits (step 1). The licensing system 23 on the application server 21 fetches the application server's server profile from the directory (not shown) using the directory API. It also fetches the directory server's server profile (step 2).

The Application server's licensing subsystem 23 analyses the licences in the server's own profile. In this case there are none, since the example is concerned with licence sharing. The licensing subsystem 23 then analyses the directory server's licences. Because there are unlimited session and usage tickets in a shareable licence, the local limit is also set to unlimited. The user ticket limit is set to 0, because they are limited (10 according to ticket database 28) and limited tickets cannot be shared with this method (step 3).

The application service 22 then requests an application session ticket from its licensing subsystem 23 (step 4). The ticket is granted because there are an unlimited supply of them. The used ticket count is updated in the local ticket database 24 (step 5), although it is only needed for statistics as the number is unlimited. The session ticket is then issued to the application service 22, which then proceeds since it has determined that there exists an appropriate licence.

License sharing in the case of a site licensing service will now be described with reference to Figure 4, in which an application server 31 provides an application service 32 and includes a licensing subsystem 33 with an associated ticket

database 34. A license server 35 provides a site licensing service 36 and includes a licensing subsystem 37 with an associated ticket database 38.

The licensing subsystems 33 and 27 of the servers 31 and 35, fetch their corresponding server profiles from a directory (not shown), analyse installed licences and store the ticket limits in the local databases 34 and 38 (step 1). The application server 31 need not have any licences.

The application service 32 requests a service ticket, for example an application session ticket, from the local licensing subsystem 33 (step 2). The local licensing subsystem 33 in the application server 31 will first attempt to issue the ticket locally, but this will fail as there are no licences installed for the application server 31, as indicated by the lack of available tickets in the ticket database 34 (step 3). The licensing subsystem 33 in the application server 31 will then connect to the site licensing service 36 using the client-server protocol and request the ticket remotely (step 4). The site licensing service 36 requests the ticket from the local licensing subsystem 37 and it also request a ticket-forwarding ticket (step 5). The licensing subsystem 37 of the license server 35 checks ticket availability and updates the used ticket counts in the ticket database 38 (step 6). The tickets are issued to the site licensing service 36 (step 7) which forwards the application ticket to the client ie licensing subsystem 33 (step 8), which as a result issues the application ticket to the application service 32, allowing that to proceed (step 9).

Whenever a licensing subsystem issues a service ticket, or a ticket is returned such as because it is an unused ticket (any number can be requested) or because it is a session ticket, which are required to be returned at the end of a session, the transaction can, optionally, be logged to a log file which is separate from other log files in the system.

The information in this separate log file may be used to implement a pay-by-usage licensing scheme (delayed billing). Logging can be enabled or disabled by an administrator. Each server has its own log file and all kinds of tickets issued in the server will be logged the same way. Logging parameters for each kind of ticket could be specified for certain types of licences, although such a licence could not be shared by the independent sharing method.

The proposed licensing method allows for introducing new services while retaining compatibility with old licences. The licensing subsystems will initially support some types of licences and service tickets that are not yet connected to any particular service. New services can be assigned to these items without making any modifications to existing administration programs and the licensing subsystem. The method could be extended further by adding new license/ticket combinations to the licensing subsystem, although all existing combinations would need to be kept unchanged. This would involve updating the licensing subsystem in all servers where the new services would be used. Older subsystems would not accept the new kind of licenses not issue tickets for the new services. The licenses and tickets could be defined statically, as they are now, although there could be other possibilities.

As discussed above, the covert code specifies the licence duration, licence size and included services. An example of a covert code comprises a 7-digit decimal number, with the digits numbered from right to left, starting from zero eg in number 6543210, digit no 0 is "0", digit no 1 is "1" etc.

Licence duration may be encoded in the last digit ie digit 0, as follows:

Digit No 0	Licence Duration
"0"	10 days
"1"	1 month (31 days)
"2"	3 months (92 days)
"3"	6 months (184 days)
"4"	1 year (366 days)
"5"	2 years
"6"	3 years
"7"	Unlimited (small size)
"8"	Unlimited (medium size)
"9"	Unlimited (large size)

Licence size may be coded in the next-to-last digit, digit no 1. However, its interpretation may depend on the licence duration. Limited duration licences may be one of, for example, 30 different sizes; duration digits "7", "8" or "9" select small, medium or large licence sizes respectively.

Digit No 1	Licence size for each duration type			
	Limited	Unlimited Small	Unlimited Medium	Unlimited Large
"0"	1	1	60	400
"1"	2	2	80	500
"2"	5	5	100	600
"3"	10	10	125	800
"4"	15	15	150	1000
"5"	20	20	175	1200
"6"	30	25	200	1500
"7"	50	30	225	2000
"8"	100	40	250	3000
"9"	Unlimited	50	300	Unlimited

The services that are included in a licence may be encoded into four digits, digits no 2 to no 5, of the covert

code. These digits are called the service code. The licence may apply to one kind of service tickets only, to a group of related service tickets that are used by one service, or to a group of selected services. The service code can be chosen to represent a particular name of service, such as "basic directory service", "basic mail service", "basic calendar service", in any desired manner but it will indicate what types of tickets are included and how many licence service tickets are included for each type of service.

The digit no 6, the most significant digit, may be used to specify a particular product line. In the examples shown in the drawings the covert codes all commence with the number 2, indicating they relate to the same product line.

Any number of licences may be installed in the server profile of any server. The activation key is verified, and the covert code calculated from the serial number and the activation key at license installation time. The mapping of covert code to service ticket is, preferably, not stored in the directory, rather it is recalculated by a licensing subsystem every time it starts up. All tickets of the same type are indistinguishable. The licensing subsystems do not keep track of individual tickets issued.

Any number of identical tickets may be obtained at once by a service from the corresponding licensing subsystem, providing of course that they are available. Tickets can be returned if they are not used.

The licensing subsystem does not force services to obtain tickets rather it is the service's responsibility to offer services only to legal users and without obtaining a respective ticket, a service which requires a licence will not function.

Session tickets are associated with client-server sessions.

Unless a service wants to allow unlimited usage, it should obtain a session ticket whenever a session starts. Determining when each session starts and ends is the responsibility of the service. Session tickets may not be applicable to all services. It is important that session tickets are returned when the sessions end, otherwise they will be unusable, at least until the licencing subsystem is resynchronised. This is achieved at server start up, when there are no sessions in existence, by setting the used session ticket count to zero.

When a user is given the right to use a service, the associated user ticket should be obtained first. Because in a currently preferred embodiment, users are created and user rights given by the directory service, the licenses that include user tickets should be installed into the directory server. The directory service is the only service that requests user tickets and it is responsible for maintaining consistency of the used ticket counts. It periodically counts all users in the directory and their user rights and sets the number of tickets in use as appropriate.

Some kinds of tickets are "consumable" e.g. for sending mail messages, and these will not be returned unless, for example, the message is cancelled.

Clearly if an originally purchased licence becomes inadequate, due for example to an increased number of users, then supplemental licences can be purchased which when installed will increase the number of available tickets for a service. Additional functionality can of course also be purchased subsequently, in order to add new features to a system, and the appropriate software and licence installed in an appropriate server.

It is considered that with the above description of the licence management system and method proposed by the

invention, a software developer will have difficulty producing the corresponding code for licence management for a particular software product written in a particular language, and hence no further description is considered necessary in this respect.

CLAIMS

1. A software licence management method for use with a computer system including at least one server, the method being such that before a service can offer functionality to a user, the said service shall verify that the user has a licence for said service, and wherein the computer system further includes a licensing subsystem with which are associated service tickets corresponding to existing licences, the method including the steps of the said service requesting a respective service ticket from the licensing subsystem prior to offering functionality to the user, and the licensing subsystem issuing a said service ticket to the said service, if one is available, thereby verifying the licence exists and allowing the said service to offer functionality.
2. A method as claimed in Claim 1, including the step of installing licence information comprising a licence serial number and a licence activation key into the computer system, the activation key containing encoded details of the licensed services, and wherein the computer system calculates, from the serial number and the activation key, information including the types of service tickets associated with a particular licence, the numbers of service tickets, and the duration of the licence.
3. A method as claimed in Claim 2 wherein the licensing subsystem maintains a log of the numbers of the maximum available and issued service tickets.
4. A method as claimed in Claim 2 or Claim 3, wherein a covert code is calculated by the computer system from the serial number and activation key and wherein mapping of the covert code to service tickets is calculated by

the licencing subsystem each time it is started.

5. A method as claimed in any one of the preceding claims wherein the computer system comprises a plurality of computers connected in a network and wherein a said server comprises a directory server, providing a directory service and including a respective licencing subsystem, together with a directory database and a ticket database, wherein stored in the directory database are directory server profile details, licence details and user profile details, and wherein the ticket database includes details of service tickets available in accordance with the respective licence details and issued, and wherein adding a user to the computer system includes the steps of starting the directory server licencing subsystem, the directory server licencing subsystem fetching the directory server profile with licence details from the directory database and updating the ticket database, the requesting of a user service ticket by the directory service from the licencing subsystem, the checking of ticket availability in the ticket database by the licencing subsystem, the issuing of a ticket by the licencing subsystem to the directory service, and the adding to the directory database of the new user's profile by the directory service.

6. A method as claimed in any one of Claims 1 to 4 wherein the computer system comprises a plurality of computers connected in a network, wherein a said server comprises an application server providing an application service and including a respective licencing subsystem with a respective ticket database, and another said server comprises a directory server providing a directory service and with a respective directory database, wherein stored in the directory database are directory server profile details, application server profile details and licence details, and wherein the ticket

database includes details of service tickets available in accordance with the licence details and issued, and wherein obtaining a use ticket for the application service includes the steps of starting the application server licensing subsystem, the subsystem fetching the application server profile and licence details from the directory database and updating the ticket database accordingly, starting the application service without providing functionality, the requesting by the application service of a user service ticket from the licensing subsystem, the checking of ticket availability in the ticket database by the licensing subsystem, and the issuing of a service ticket to the application service by the licensing subsystem, the application service then providing its functionality to a user.

7. A method as claimed in any one of Claims 1 to 4 and for independent licence sharing, wherein the computer system comprises a plurality of computers connected in a network, wherein a said server comprises an application server providing an application service and including a respective licensing subsystem with a respective ticket database, and another said server comprises a directory server providing a directory service and including a respective licensing subsystem with a respective ticket base and with a respective directory database, wherein stored in the directory database are directory server profile details, application server profile details and shareable licence details, the number of service tickets being unlimited, wherein the directory server ticket database includes details of service tickets available in accordance with the shareable licence details and issued, and wherein the application server ticket database includes details of service tickets issued, and wherein obtaining a service ticket for the application service includes the steps of the directory server licensing system fetching the server profile from the

directory database, analysing the shareable licence details and updating the corresponding ticket types and ticket limits in the directory server ticket database, the application server licensing subsystem fetching the application server and the directory server profiles and shareable licence details from the directory database and analysing them and updating the corresponding ticket types in the application server ticket database, starting the application service without providing functionality, the requesting by the application service of a service ticket from the application server licensing system, the granting of a service ticket, and the issuing of the service ticket to the application service by the application server licensing system, the application service then providing its functionality to a user.

8. A method as claimed in any one of Claims 1 to 4 and for licence sharing with site licensing, wherein the computer system comprises a plurality of computers connected in a network, wherein a said server comprises an application server providing an application service and including a respective licensing subsystem with a respective ticket database, another said server comprises a site licensing server providing a site licensing service and including a respective licensing subsystem with a respective ticket database, and a further said server comprises a directory server providing a directory service and having a directory database, wherein stored in the directory database are directory server profile details, site licensing server profile details, application server profile details and licence details, wherein the site licensing subsystem ticket database includes details of service tickets available in accordance with the licence details and issued, and wherein obtaining a service ticket for the application service when the application server has no

respective licence includes the steps of the licensing subsystems fetching their corresponding server profiles from the directory database, analysing the installed licence details and the site licensing server updating the respective ticket database, starting the application service without providing functionality, the requesting by the application service of a service ticket from the site licensing service, the requesting of a service ticket and a ticket-forwarding ticket by the site licensing service from its licensing subsystem, the checking of ticket availability and the issuing of the service and ticket-forwarding tickets to the site licensing service, the forwarding of the service ticket to the application server licensing subsystem, and the issuing of the service ticket to the application service, the application service then providing its functionality to a user.

9. A computer system including at least one server and a software licence management system, the management system being such that before a service can offer functionality to a user, the service shall verify that the user has a licence for said service, the management system including a licencing subsystem with which are associated service tickets corresponding to existing licences, and the management system being such that a said service ticket is issued to a service, if one is available, upon request by the service, thereby verifying existence of a licence and allowing the said service to offer functionality.

10. A computer system as claimed in Claim 9, wherein the management system includes means for calculating from an input licence serial number and input licence activation key, information including the types of service tickets associated with a particular licence, the numbers of service tickets and the duration of the licence, said

information being encoded in the activation key.

11. A computer system as claimed in Claim 10, and including a log in which are stored the numbers of the maximum available and issued service tickets.
12. A computer system as claimed in Claim 9 or Claim 10, and wherein the calculating means include means for calculating a covert code from the serial number and activation key, and the licensing subsystem including means for mapping the covert code into service tickets each time the licensing subsystem is started.
13. A computer system as claimed in any one of Claims 9 to 12 and comprising a plurality of computers connected in a network, wherein a said server comprises a directory server, providing a directory service and including a respective licensing subsystem, together with a directory database and a ticket database, wherein stored in the directory database are directory server profile details, licence details and user profile details, and wherein the ticket database includes details of service tickets available in accordance with respective licence details and issued.
14. A computer system as claimed in any one of Claims 9 to 12 and comprising a plurality of computers connected in a network, wherein a said server comprises an application server providing an application service and including a respective licensing subsystem with a respective ticket database, and another server comprises a directory server providing a directory service with a respective directory database, wherein stored in the directory database are directory server profile details, application server profile details and licence details, and wherein the ticket database includes details of service tickets available in accordance with the licence

details and issued.

15. A computer system as claimed in Claim 14 and for independent licence sharing, wherein the directory server includes a respective directory licensing subsystem and a respective ticket database, shareable licence details, for which the number of service tickets available is unlimited, being stored in the directory database, the directory ticket database including details of service tickets available in accordance with the shareable licence details and issued, and the application server ticket database including details of service tickets issued.
16. A computer system as claimed in Claim 14 and for licence sharing with site licensing, and including another said server comprising a site licensing server providing a site licensing service and including a respective licensing subsystem with a respective ticket database, the directory database also including site licensing server profile details, and wherein the site licensing ticket database includes details of service tickets available in accordance with the licence detailed and issued.
17. A software licence management method substantially as herein described with reference to an as illustrated in Figure 1, Figure 2, Figure 3, or Figure 4, of the accompanying drawings.
18. A computer system including at least one server and a software licence management system substantially as herein described with reference to and as illustrated in Figure 1, or Figure 2, or Figure 3, or Figure 4 of the accompanying drawings.



The
Patent
Office

29

Application No: GB 9617596.3
Claims searched: 1-18

Examiner: Mike Davis
Date of search: 26 September 1996

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): G4A (AAP)

Int Cl (Ed.6): G06F

Other:

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X	GB 2236604 A (SUN MICROSYSTEMS)	1,9 at least
X	EP 0332304 A2 (DIGITAL EQUIPMENT)	.
X	WO 93/11480 A1 (INTERGRAPH)	.
X	US 5375206 (HUNTER ET AL)	.
X	US 4924378 (HERSHEY ET AL)	.

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

An Executive Agency of the Department of Trade and Industry

E26 1 PN=BR 9810991
 E27 1 PN=BR 9810992
 E28 1 PN=BR 9810993
 E29 1 PN=BR 9810994
 E30 1 PN=BR 9810995
 E31 1 PN=BR 9810996
 E32 1 PN=BR 9810997
 E33 1 PN=BR 9810998
 E34 1 PN=BR 9810999
 E35 1 PN=BR 9811000
 E36 1 PN=BR 9811001
 E37 1 PN=BR 9811002
 E38 1 PN=BR 9811004
 E39 1 PN=BR 9811005
 E40 1 PN=BR 9811006
 E41 1 PN=BR 9811007
 E42 1 PN=BR 9811008
 E43 1 PN=BR 9811009
 E44 1 PN=BR 9811010
 E45 1 PN=BR 9811011
 E46 1 PN=BR 9811012
 E47 1 PN=BR 9811013
 E48 1 PN=BR 9811014
 E49 1 PN=BR 9811015
 E50 1 PN=BR 9811016

Enter P or PAGE for more

? s e3

S1 1 PN='BR 9810967'

? t 1/7/1

1/7/1

DIALOG(R)File 351: Derwent WPI

(c) 2008 The Thomson Corporation. All rights reserved.

0009253575 *Drawing available*

WPI Acc no: 1999-181268/199915

Related WPI Acc No: 1996-465320; 1997-363998; 1998-363180; 1999-154174; 1999-154175;
 1999-154176; 1999-154177; 1999-154178; 1999-154179; 1999-243551; 2002-060946; 2002-
 499082; 2002-705909; 2002-722051; 2002-722052; 2003-677663; 2003-898213; 2004-155029;
 2004-478232; 2004-579235; 2004-623798; 2005-809338; 2007-015228

XRPX Acc No: N1999-133079

method for decrypting an instance of service that has been decrypted with short-term key

Patent Assignee: SCIENTIFIC-ATLANTA INC (SCAT)

Inventor: AKINS G L; PALGON M S; PINDER H G; WASILEWSKI A J; AKINS G

Patent Family (8 patents, 79 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	Type
WO 1999009743	A2	19990225	WO 1998US16079	A	19980731	199915	B

AU 199915816	A	19990308	AU 199915816	A	19980731	199929	E
EP 1000511	A2	20000517	EP 1998960147	A	19980731	200028	E
			WO 1998US16079	A	19980731		
BR 199810967	A	20011030	BR 199810967	A	19980731	200173	E
			WO 1998US16079	A	19980731		
EP 1000511	B1	20011114	EP 1998960147	A	19980731	200175	E
			WO 1998US16079	A	19980731		
DE 69802540	E	20011220	DE 69802540	A	19980731	200207	E
			EP 1998960147	A	19980731		
			WO 1998US16079	A	19980731		
JP 2003521820	W	20030715	WO 1998US16079	A	19980731	200347	E
			JP 2000510276	A	19980731		
JP 2005253109	A	20050915	JP 2000510276	A	19980731	200560	E
			JP 2005120425	A	20050418		

Priority Applications (no., kind, date): US 199754575 P 19970801; US 1998126921 A 19980731

Patent Details

Patent Number	Kind	Lan	Pgs	Draw	Filing Notes	
WO 1999009743	A2	EN	113	29		
National Designated States, Original	AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM HR HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZW					
Regional Designated States, Original	AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW					
AU 199915816	A	EN			Based on OPI patent	WO 1999009743
EP 1000511	A2	EN			PCT Application	WO 1998US16079
					Based on OPI patent	WO 1999009743
Regional Designated States, Original	DE FR GB IT NL					
BR 199810967	A	PT			PCT Application	WO 1998US16079
					Based on OPI patent	WO 1999009743
EP 1000511	B1	EN			PCT Application	WO 1998US16079
					Based on OPI patent	WO 1999009743
Regional Designated States, Original	DE FR GB IT NL					
DE 69802540	E	DE			Application	EP 1998960147
					PCT Application	WO 1998US16079
					Based on OPI patent	EP 1000511

JP 2003521820	W	JA	136	Based on OPI patent	WO 1999009743
				PCT Application	WO 1998US16079
JP 2005253109	A	JA	59	Based on OPI patent	WO 1999009743
				Division of application	JP 2000510276

Alerting Abstract WO A2

NOVELTY - The method involves receiving a second message in a receiver together with the instance of the service. The second message includes a key derivation value that is used with a long-term key to obtain the short-term key to decrypt the instance of the service.

DESCRIPTION - A control word is combined into an encrypted coded message (ECM) (107) with other service-related information. The ECM (107) is authenticated by Control Word Encrypt & Message Authenticate function (204) which produces a message authentication code using a keyed-hash value derived from the message content combined with a secret which can be shared with the receiving set-top box (113). This secret is preferably part or all of a multisession key (MSS) (208). The message authentication code is appended to the rest of the ECM (107). The CAW (202) is always encrypted before being sent along with the other parts of the ECM to MX (200). This encryption is preferably a symmetric cipher such as the Triple-DES algorithm using two distinct 56-bit keys (which taken together comprise MSS (208)).

USE - The invention concerns systems for protecting information and more particularly concerns systems for protecting information that is transmitted by a wired or wireless medium against unauthorized access.

ADVANTAGE - The service distribution organizations require access restrictions which are both more secure and more flexible than those in conventional systems

DESCRIPTION OF DRAWINGS - The drawing is a block diagram of service instance encryption techniques.

107 encrypted coded message

204 Control Word Encrypt & Message Authenticate function

200 MX

Title Terms /Index Terms/Additional Words: METHOD; INSTANCE; SERVICE; SHORT; TERM; KEY

Class Codes

International Patent Classification

IPC	Class Level	Scope	Position	Status	Version Date
H04L-009/08			Main		"Version 7"
H04H-001/00; H04N-007/167; H04N-007/173			Secondary		"Version 7"
H04H-0001/00	A	I	L	R	20060101
H04L-0009/08	A	I	L	R	20060101
H04N-0005/00	A	I		R	20060101
H04N-0007/16	A	I		R	20060101
H04N-0007/167	A	I		R	20060101
H04N-0007/173	A	I	F	R	20060101

H04H-0001/00	C	I	L	R	20060101
H04L-0009/08	C	I	F	R	20060101
H04N-0005/00	C	I		R	20060101
H04N-0007/16	C	I		R	20060101
H04N-0007/167	C	I		R	20060101
H04N-0007/173	C	I	L	R	20060101

File Segment: EPI;
 DWPI Class: W02; W03
 Manual Codes (EPI/S-X): W02-F05A1B; W03-A16C3A

Original Publication Data by Authority

Australia

Publication No. AU 199915816 A (Update 199929 E)
Publication Date: 19990308
Assignee: SCIENTIFIC-ATLANTA INC; US (SCAT)
Language: EN
Application: AU 199915816 A 19980731 (Local application)
Priority: US 199754575 P 19970801
 US 1998126921 A 19980731
Related Publication: WO 1999009743 A (Based on OPI patent)
Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00
 (R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08
 (R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00
 (R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16
 (R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167
 (R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220,A,F) H04N-7/173
 (R,I,M,JP,20060101,20051220,C,L)

Brazil

Publication No. BR 199810967 A (Update 200173 E)
Publication Date: 20011030
Assignee: SCIENTIFIC-ATLANTA INC (SCAT)
Inventor: WASILEWSKI A J
 AKINS G L
 PALGON M S
 PINDER H G
Language: PT
Application: BR 199810967 A 19980731 (Local application)
 WO 1998US16079 A 19980731 (PCT Application)
Priority: US 199754575 P 19970801
 US 1998126921 A 19980731
Related Publication: WO 1999009743 A (Based on OPI patent)

Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00
 (R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08
 (R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00
 (R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16
 (R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167
 (R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220,A,F) H04N-7/173
 (R,I,M,JP,20060101,20051220,C,L)

Germany

Publication No. DE 69802540 E (Update 200207 E)
Publication Date: 20011220
Assignee: SCIENTIFIC-ATLANTA INC; US (SCAT)
Language: DE
Application: DE 69802540 A 19980731 (Local application)
 EP 1998960147 A 19980731 (Application)
 WO 1998US16079 A 19980731 (PCT Application)
Priority: US 199754575 P 19970801
 US 1998126921 A 19980731
Related Publication: EP 1000511 A (Based on OPI patent)
 WO 1999009743 A (Based on OPI patent)

EPO

Publication No. EP 1000511 A2 (Update 200028 E)
Publication Date: 20000517
Assignee: SCIENTIFIC-ATLANTA, INC., One Technology Parkway South, Norcross, Georgia 30092, US
Inventor: AKINS, Glendon, L., III, 2510 Windward Lane N.E., Gainesville, GA 30501, US
 PALGON, Michael, S., 1196 Poplar Grove Drive, Atlanta, GA 30306, US
 PINDER, Howard, G., 4317 Stilson Circle, Norcross, GA 30092, US
 WASILEWSKI, Anthony, J., 10680 Wren Ridge Road, Alpharetta, GA 30022, US
Agent: Kugele, Bernhard, NOVAPAT INTERNATIONAL SA, 9, Rue du Valais, 1202 Geneve, CH
Language: EN
Application: EP 1998960147 A 19980731 (Local application)
 WO 1998US16079 A 19980731 (PCT Application)
Priority: US 199754575 P 19970801
 US 1998126921 A 19980731
Related Publication: WO 1999009743 A (Based on OPI patent)
Designated States: (Regional Original) DE FR GB IT NL
Original IPC: H04N-7/167(A)
Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00
 (R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08
 (R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00
 (R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16
 (R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167
 (R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220,A,F) H04N-7/173
 (R,I,M,JP,20060101,20051220,C,L)
Original Abstract:

A cable television system provides conditional access to services. The cable television system includes a headend from which service "instances", or programs, are broadcast and a plurality of set top units for receiving the instances and selectively decrypting the instances for display to system subscribers. The service instances are encrypted using public and/or private keys provided by service providers or central authorization agents. Keys used by the set tops for selective decryption may also be public or private in nature, and such keys may be reassigned at different times to provide a cable television system in which piracy concerns are minimized.

Publication No. EP 1000511 B1 (Update 200175 E)

Publication Date: 20011114

Assignee: Scientific-Atlanta, Inc., 5030 Sugarloaf Parkway, Lawrenceville, GA 30044, US

Inventor: AKINS, Glendon, L., III, 2510 Windward Lane N.E., Gainesville, GA 30501, US

PALGON, Michael, S., 1196 Poplar Grove Drive, Atlanta, GA 30306, US

PINDER, Howard, G., 4317 Stilson Circle, Norcross, GA 30092, US

WASILEWSKI, Anthony, J., 10680 Wren Ridge Road, Alpharetta, GA 30022, US

Agent: Kugele, Bernhard, NOVAPAT INTERNATIONAL SA, 9, Rue du Valais, 1202 Geneve, CH

Language: EN

Application: EP 1998960147 A 19980731 (Local application)

WO 1998US16079 A 19980731 (PCT Application)

Priority: US 199754575 P 19970801

US 1998126921 A 19980731

Related Publication: WO 1999009743 A (Based on OPI patent)

Designated States: (Regional Original) DE FR GB IT NL

Original IPC: H04N-7/167(A)

Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00

(R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08

(R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00

(R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16

(R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167

(R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220,A,F) H04N-7/173

(R,I,M,JP,20060101,20051220,C,L)

Claim:

1. Verfahren der Entschlüsselung einer Diensteeinheit (325), die mit einem gegebenen Kurzzeitschlüssel (319) verschlüsselt wurde, wobei das Verfahren in einem Empfänger (333) ausgeführt wird, der ein Öffentlich/Privat-Schlüsselpaar besitzt, und das Verfahren durch die folgenden Schritte **gekennzeichnet** ist:
 - o im Empfänger eine erste Nachricht (315) zu empfangen, deren Inhalt einen ersten Langzeitschlüssel (309) einschliesst und unter Verwendung des öffentlichen Schlüssels (312) für den Empfänger (333) verschlüsselt wurde;
 - o den privaten Schlüssel (337) zur Entschlüsselung des Inhalts zu verwenden;
 - o den ersten Schlüssel (309) zu speichern;
 - o im Empfänger (333) zusammen mit der verschlüsselten Diensteeinheit (329) eine zweite Nachricht (323) zu empfangen, wobei die zweite Nachricht (323) einen Indikator für einen zweiten Kurzzeitschlüssel (319) einschliesst;
 - o den Indikator und den ersten Schlüssel (309) zu benutzen, um den zweiten Schlüssel zu erhalten; worin der zweite Schlüssel dem gegebenen Schlüssel (319), mit dem der Dienst verschlüsselt wurde, gleichwertig ist, und
 - o den zweiten Schlüssel zur Entschlüsselung der empfangenen Diensteeinheit zu

verwenden.

1. A method of decrypting an instance of a service (325) that has been encrypted with a given short-term key (319), the method being carried out in a receiver (333) that has a public key-private key pair and the method being **characterised by** the following steps:
 - o receiving a first message (315) in the receiver whose contents include a first long-term key (309), the contents having been encrypted using the public key (312) for the receiver (333);
 - o using the private key (337) to decrypt the contents;
 - o storing the first key (309);
 - o receiving a second message (323) in the receiver (333) together with the encrypted instance of the service (329), the second message (323) including an indicator for a second short-term key (319);
 - o using the indicator and the first key (309) to obtain the second key; wherein the second key is equivalent to the given key (319) that encrypted the service, and
 - o using the second key to decrypt the received instance of the service.

1. Procéde de decryptage d'une instance d'un service (326) qui était cryptée avec une cle a court terme donnée (319), le procéde étant exécuté dans un récepteur (333) qui comporte une paire de cle publique-cle privée et le procéde étant **caractérisé par** les étapes suivantes:
 - o recevoir un premier message (315) dans le récepteur dont le contenu comprend une première cle a long terme (309), le contenu ayant été crypté en utilisant la cle publique (312) pour le récepteur (333),
 - o utiliser la cle privée (337) pour decrypter le contenu,
 - o mémoriser la première cle (309),
 - o recevoir un second message (323) dans le récepteur (333) en même temps que l'instance cryptée du service (329), le second message (323) comprenant un indicateur pour une seconde cle a court terme (319),
 - o utiliser l'indicateur et la première cle (309) pour obtenir la seconde cle, dans lequel
 - o la seconde cle est équivalente a la cle donnée (319) qui a crypté le service, et
 - o utiliser la seconde cle pour decrypter l'instance reçue du service.

Japan

Publication No. JP 2003521820 W (Update 200347 E)

Publication Date: 20030715

Language: JA (136 pages)

Application: WO 1998US16079 A 19980731 (PCT Application)

JP 2000510276 A 19980731 (Local application)

Priority: US 199754575 P 19970801

US 1998126921 A 19980731

Related Publication: WO 1999009743 A (Based on OPI patent)

Original IPC: H04L-9/08(A) H04H-1/00(B) H04N-7/167(B) H04N-7/173(B)
 Current IPC: H04L-9/08(A) H04H-1/00(B) H04N-7/167(B) H04N-7/173(B)

Publication No. JP 2005253109 A (Update 200560 E)

Publication Date: 20050915

CONDITIONAL ACCESS SYSTEM

Assignee: SCIENTIFIC-ATLANTA INC (SCAT)

Inventor: AKINS GLENDON L III

PALGON MICHAEL S

PINDER HOWARD G

WASILEWSKI ANTHONY J

Language: JA (59 pages)

Application: JP 2000510276 A 19980731 (Division of application)

JP 2005120425 A 20050418 (Local application)

Priority: US 199754575 P 19970801

US 1998126921 A 19980731

Original IPC: H04L-9/08(A)

Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00

(R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08

(R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00

(R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16

(R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167

(R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220,A,F) H04N-7/173

(R,I,M,JP,20060101,20051220,C,L)

WIPO

Publication No. WO 1999009743 A2 (Update 199915 B)

Publication Date: 19990225

CONDITIONAL ACCESS SYSTEM

RESEAU D'ACCES CONDITIONNEL

Assignee: SCIENTIFIC-ATLANTA, INC., Intellectual Property Dept., One Technology Parkway South, Norcross, GA 30092, US Residence: US Nationality: US (SCAT)

Inventor: AKINS, Glendon, L., III, 2510 Windward Lane N.E., Gainesville, GA 30501, US

PALGON, Michael, S., 1196 Poplar Grove Drive, Atlanta, GA 30306, US

PINDER, Howard, G., 4317 Stilson Circle, Norcross, GA 30092, US

WASILEWSKI, Anthony, J., 10680 Wren Ridge Road, Alpharetta, GA 30022, US

Agent: GARDNER, Kelly, A., Scientific-Atlantic, Inc., Intellectual Property Dept., One Technology Parkway South, Norcross, GA 30092, US

Language: EN (113 pages, 29 drawings)

Application: WO 1998US16079 A 19980731 (Local application)

Priority: US 199754575 P 19970801

US 1998126921 A 19980731

Designated States: (National Original) AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM HR HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZW

(Regional Original) AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW

Original IPC: H04N-7/167(A)

Current IPC: H04H-1/00(R,I,M,JP,20060101,20051220,A,L) H04H-1/00

(R,I,M,JP,20060101,20051220,C,L) H04L-9/08(R,I,M,JP,20060101,20051220,A,L) H04L-9/08
 (R,I,M,JP,20060101,20060310,C,F) H04N-5/00(R,I,M,EP,20060101,20051008,A) H04N-5/00
 (R,I,M,EP,20060101,20051008,C) H04N-7/16(R,I,M,EP,20060101,20051008,A) H04N-7/16
 (R,I,M,EP,20060101,20051008,C) H04N-7/167(R,I,M,EP,20060101,20051008,A) H04N-7/167
 (R,I,M,EP,20060101,20051008,C) H04N-7/173(R,I,M,JP,20060101,20051220, A,F) H04N-7/173
 (R,I,M,JP,20060101,20051220,C,L)

Original Abstract:

A cable television system provides conditional access to services. The cable television system includes a headend from which service "instances", or programs, are broadcast and a plurality of set top units for receiving the instances and selectively decrypting the instances for display to system subscribers. The service instances are encrypted using public and/or private keys provided by service providers or central authorization agents. Keys used by the set tops for selective decryption may also be public or private in nature, and such keys may be reassigned at different times to provide a cable television system in which piracy concerns are minimized.

Un reseau de television par cable assure un acces conditionnel a des services. Le reseau de television par cable comprend une tete de reseau a partir de laquelle on diffuse les "instances" de service ou programmes. Ce reseau comprend aussi une pluralite d'unites decodeurs concues pour recevoir les instances et dechiffrer selectivement les instances qui vont s'afficher pour les abonnes du reseau. Les instances de service sont chiffrees par des cles publiques et/ou privees fournies par des fournisseurs de service ou des agents d'autorisation centraux. Les cles utilisees par les decodeurs permettant un dechiffrement selectif peuvent aussi etre publiques ou privees et de telles cles peuvent etre reffectees a differents moments pour assurer un reseau de television par cable dans lequel les risques de piratage sont minimisees.

?



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
 11.08.1999 Bulletin 1999/32

(51) Int. Cl.⁶: **A63F 9/22**

(21) Application number: **98400285.7**

(22) Date of filing: **09.02.1998**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
 Designated Extension States:
AL LT LV MK RO SI

(72) Inventors:
 • **Agasse, Bernard**
95610 Eragny/Oise (FR)
 • **Bayassi, Mulham**
75015 Paris (FR)

(60) Divisional application:
98202314.5

(74) Representative:
Cozens, Paul Dennis et al
Mathys & Squire
100 Grays Inn Road
London WC1X 8AL (GB)

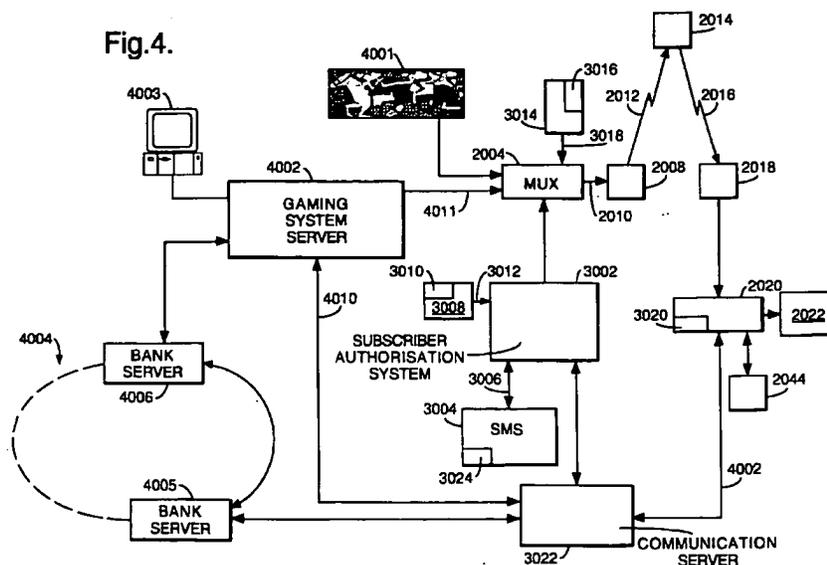
(71) Applicant:
CANAL+ Société Anonyme
75711 Paris Cedex 15 (FR)

Remarks:
 The application is published incomplete as filed (Article 93 (2) EPC).

(54) **Interactive gaming system**

(57) An interactive gaming and audiovisual transmission system comprising a central gaming computer 4002 for processing gaming data, a decoder 2020 adapted to receive gaming data from the central gaming computer 4002 together with transmitted audiovisual data, the decoder further including a card reading

device for interacting with a user's bank card in order to credit a gaming account held by the central gaming computer means in response to a transfer of credit from the user's bank account.



EP 0 934 765 A1

Description

[0001] The present invention relates to an interactive gaming and digital audiovisual transmission system, in particular a gaming and digital television transmission system.

[0002] Broadcast transmission of digital data is well-known in the field of pay TV systems, where scrambled audiovisual information is sent, usually by a satellite or satellite/cable link, to a number of subscribers, each possessing a decoder capable of descrambling the transmitted program for subsequent viewing. Terrestrial digital broadcast systems are also known. Recent systems have also used the broadcast link to transmit other data, in addition to or as well as audiovisual data, such as computer programs or interactive applications to the decoder or to a connected PC.

[0003] The increasing sophistication of such technology, in particular in relation to the receiver/decoder devices used in the systems, has led to an increase in the possible services that may be provided thereby. In particular, a number of systems have been proposed using interactive technology to enable a viewer to, for example, participate in a quiz show, or to select further information regarding a product currently being displayed on a shopping channel.

[0004] In the case of gaming applications, a number of largely theoretical systems have been proposed to enable a viewer to gamble a sum of money on the outcome of a sporting event or casino-type game broadcast over a television network. In most of these systems, a viewer is usually obliged to open an initial account with the controlling gaming authority by phoning or mailing a money transfer to the gaming authority before any gambling can be carried out. The disadvantages of this sort of procedure will be apparent.

[0005] Alternative systems are also known, in which the viewer buys credits to be gambled in the form of an electronic purse, i.e. a smart card or the like, the credits in the purse being available for subsequent gaming operations. The card is inserted in the decoder and the credits used thereafter in the subsequent gaming operations. When the contents of the purse are exhausted, the viewer buys a new card or re-charges the card at a suitable sales point. This system again implies a certain infra-structure to be put in place to enable a user to obtain the necessary credits to be gambled.

[0006] The present invention seeks to overcome some or all of the disadvantages of these prior art systems.

[0007] According to the present invention, there is provided an interactive gaming and audiovisual transmission system comprising a central gaming computer means for processing gaming data, a decoder adapted to receive gaming data from the central gaming computer together with transmitted audiovisual data, the decoder further including a card reading device for interacting with a user's bank card in order to credit a gaming account held by the central gaming computer means

in response to a transfer of credit from the user's bank account.

[0008] In this way, the present invention enables a user to simply and quickly open and credit a gaming account from the comfort of his home, avoiding the more elaborate payment methods of the known systems.

[0009] The type of bank card used in this transaction may be of the debit or credit type. The card reading device may in particular comprise a smart card reader adapted to interact with a bank card in the form of a smart card.

[0010] Advantageously, the decoder is further equipped with a second card reading device. For example, in the case where the decoder forms part of a television subscription service, the subscriber may be provided with a subscription card in the form of a smart card or the like. The provision of two card reader devices in the decoder permits the decoder to carry out credit transactions on a bank card inserted in one reader whilst the subscription card is held in the second reader.

[0011] In one realisation, the decoder may be adapted to obtain transfer of credit information in the form of an electronic certificate generated by the bank card in response to transaction data submitted by the decoder. This transaction information may include, for example, the details of the bank account of the gaming authority to be credited in the operation, the sum of money to be transferred etc.

[0012] Typically, data is entered by the user into the decoder using a handheld remote control. In the case where a credit transaction is to be carried out, it may be necessary to enter the bank card PIN number using the remote control. In one embodiment, the decoder is provided with a handheld remote control, some or all of the data sent to the decoder being encrypted by the handheld remote control and subsequently decrypted by the decoder. In this way, interception by third parties of sensitive data emitted by the remote control may be avoided.

[0013] Preferably, the decoder is adapted to transmit transfer of credit information from the decoder to a bank server via a network communication link, for example, using an modem integrated in the decoder.

[0014] The decoder may be adapted to directly communicate transfer of credit information to a bank computer. However, preferably, the system further comprises an intermediate communications server, adapted to receive transfer of credit information communicated from the decoder and to forward this information on to a bank server.

[0015] The intermediate communications server may further be adapted to communicate with the central gaming computer means, for example, to inform the central communication means of a transfer of credit instruction being forwarded from the intermediate communication means to a bank computer, so as to permit

the gaming computer means to set up an account without having to verify the transaction carried out at an associated bank server.

[0016] The central gaming computer means may equally be adapted to receive and transmit credit information to or from a bank server via a network communication link. This may be necessary, for example, in the case of a win or in order to verify the transfer of funds from the bank account of a user to the gaming authorities bank account before opening a gaming account.

[0017] Preferably, the decoder is adapted to communicate gaming information to the central gaming computer during gaming operation via a network communication link. This may be the same link as used to communicate transfer of credit information to a bank computer, for example, using a modem device integrated in the decoder.

[0018] Some or all of the gaming information communicated from the decoder to the central gaming computer during gaming operation may be encrypted by the decoder. For example, the decoder may be adapted to transmit in encrypted form a code word entered by the user associated with the gaming account of the user held by the central gaming computer.

[0019] The decoder may be adapted to directly communicate information to the central gaming computer during gaming operation. However, preferably, the system further comprises an intermediate communications server, adapted to receive information communicated from the decoder during gaming operation and to forward this information on to the central gaming computer. This may be the same intermediate server as used for the transfer of credit information between the decoder and a bank.

[0020] In the case where gaming information is encrypted by the decoder, the intermediate communications server may be adapted to simply pass this information "as is" to the central gaming computer. However, in one embodiment, the intermediate communications server is adapted to decrypt information received from the decoder and to re-encrypt this information for subsequent communication to the central gaming computer. This may be required, for example, in the case where different encryption algorithms are used by the decoder and central gaming computer.

[0021] The intermediate communications server may further be adapted to communicate information to and from other computer devices, for example, computer databases holding TV subscriber information. In this way, the intermediate communications server may obtain directly information regarding the user of the system (name, address etc) to be used in setting up a gaming account, without the user having to re-enter the same information.

[0022] The communication means used to transmit gaming data from the central gaming computer to the decoder may be defined in a number of different ways and by a number of different communication elements.

For example, some or all of the gaming data sent from the gaming computer to the decoder may be transmitted via a transmitter means used to transmit audiovisual data to the decoder.

[0023] In addition, or alternatively, some or all of the gaming data sent from the central gaming computer to the decoder may be sent via a network communication link, for example, the same network used to communicate information from the decoder to the central gaming computer during gaming operation.

[0024] In practice, a mixture of these two communication paths may prove optimal, the network path being used for rapid dialogue between the decoder and the gaming computer during real-time operation and the transmission path being used for relatively fixed data, such as screen format display data or the like.

[0025] The present invention also extends to a gaming system for processing gaming data, comprising:

- means for transmitting gaming data to a user's decoder;
- means for receiving data from the user's decoder; and
- means for connection to a bank server holding the user's bank account in order to transfer credit to or from the account.

[0026] The gaming system may include a gaming account held by the gaming system which can be credited in response to the transfer of credit.

[0027] The gaming system may be adapted to communicate with the decoder and the bank server via a communications server. If so, the gaming system may be adapted to receive encrypted information from the communications server.

[0028] The present invention also provides an interactive gaming and audiovisual transmission system comprising a gaming system as aforementioned, said user's decoder, and said bank server.

[0029] As mentioned above the system may be used to permit gaming in relation to various events. For example, the central gaming computer may be adapted to generate a computer game (computer blackjack or the like), the computer generated images being transmitted via the audiovisual link to the decoder.

[0030] However, as will be appreciated, the combination of gaming and audiovisual systems makes the present invention particularly adapted to permit gaming in relation to televised sports, such as horse racing or the like. In one embodiment, the present invention comprises a central gaming computer adapted to provide gaming data related to a real-time sporting event, the decoder being adapted to receive both gaming data and associated audiovisual data of the event.

[0031] In the context of the present application the term ((audiovisual transmission system)) refers to all transmission systems for transmitting or broadcasting primarily audiovisual or multimedia digital data. The

present invention is particularly, but not exclusively, applicable to a broadcast digital television system.

[0032] In this application the term ((smart card)) is used to mean any conventional chip-based card device possessing, for example, microprocessor and/or memory storage. Also included in this term are chip devices having alternative physical forms, for example key-shaped devices such as are often used in TV decoder systems.

[0033] In the present application, the term "decoder" is used to apply to an integrated receiver/decoder for receiving and decrypting an encrypted transmission, the receiver and decoder elements of such a system as considered separately, as well as to a receiver capable of receiving non-encrypted broadcasts. The term equally covers decoders including additional functions, such as web browsers, together with decoder systems integrated with other devices, for example, integrated VHS/decoder devices or the like.

Figure 1 shows the overall architecture of a digital television system, as may be incorporated in the gaming system of the present invention;

Figure 2 shows the conditional access system of the television system of Figure 1;

Figure 3 shows the structure of the decoder of Figures 1 and 2;

Figure 4 shows a gaming system incorporating the television system of Figures 1 and 2; and

Figure 5 shows a flow diagram of the logical steps involved in a gaming transaction

Digital Television System

[0034] An overview of a digital television broadcast and reception system 1000 adaptable to the present invention is shown in Figure 1. The system includes a mostly conventional digital television system 2000, which uses the known MPEG-2 compression system to transmit compressed digital signals. In more detail, the MPEG-2 compressor 2002 in a broadcast centre receives a digital signal stream (typically a stream of video signals). The compressor 2002 is connected to a multiplexer and scrambler 2004 by linkage 2006. The multiplexer 2004 receives a plurality of further input signals, assembles one or more transport streams and transmits compressed digital signals to a transmitter 2008 of the broadcast centre via linkage 2010, which can of course take a wide variety of forms including telecom links.

[0035] The transmitter 2008 transmits electromagnetic signals via uplink 2012 towards a satellite transponder 2014, where they are electronically processed and broadcast via notional downlink 2016 to earth

receiver 2018, conventionally in the form of a dish owned or rented by the end user. The signals received by receiver 2018 are transmitted to an integrated receiver/decoder 2020 owned or rented by the end user and connected to the end user's television 2022. The receiver/decoder 2020 decodes the compressed MPEG-2 signal into a television signal for the television set 2022.

[0036] A conditional access system 3000 is connected to the multiplexer 2004 and the receiver/decoder 2020, and is located partly in the broadcast centre and partly in the decoder. It enables the end user to access digital television broadcasts from one or more broadcast suppliers. A smart card, capable of decrypting messages relating to commercial offers (that is, on or several television programmes sold by the broadcast supplier), can be inserted into the receiver/decoder 2020. Using the decoder 2020 and smart card, the end user may purchase events in either a subscription mode or a pay-per-view-mode.

[0037] An interactive system 4000, also connected to the multiplexer 2004 and the receiver/decoder 2020 and again located partly in the broadcast and partly in the decoder, enables the end user to interact with various applications via a modemmed back channel 4002. Such interactive applications may include an interactive shopping service, a quiz application, an interactive programme guide etc.

[0038] In point of fact, whilst the interactive system 4000 has been represented as a discrete logical block, the physical elements of this system, such as the server or servers used to handle communications between the receiver/decoder and central servers, may be elements shared with the conditional access system 3000. This will become clear in the description of the gaming system of Figure 4.

Conditional Access System

[0039] With reference to Figure 2, the conditional access system 3000 includes a Subscriber Authorization System (SAS) 3002. The SAS 3002 is connected to one or more Subscriber Management Systems (SMS) 3004, one SMS for each broadcast supplier, by a respective TCP-IP link 3006 (although other types of linkage could alternatively be used). Alternatively, one SMS could be shared between two broadcast suppliers, or one supplier could use two SMSs, and so on.

[0040] First encrypting units in the form of ciphering units 3008 utilising ((mother)) smart cards 3010 are connected to the SAS by linkage 3012. Second encrypting units again in the form of ciphering units 3014 utilising mother smart cards 3016 are connected to the multiplexer 2004 by linkage 3018. The receiver/decoder 2020 receives a ((daughter)) smart card 3020. It is connected directly to the SAS 3002 by Communications Servers 3022 via the modemmed back channel 4002. The SAS sends amongst other things subscription

rights to the daughter smart card on request.

[0041] The smart cards contain the secrets of one or more commercial operators. The ((mother)) smart card encrypts different kinds of messages and the ((daughter)) smart cards decrypt the messages, if

[0042] The first and second ciphering units 3008 and 3014 comprise a rack, an electronic VME card with software stored on an EEPROM, up to 20 electronic cards and one smart card 3010 and 3016 respectively, for each electronic card, one (card 3016) for encrypting the ECMs and one (card 3010) for encrypting the EMMS.

[0043] Also shown in Figure 2 is a handheld remote control used by the viewer to control and program functions of the receiver/decoder 2020.

Multiplexer and Scrambler

[0044] With reference to Figures 1 and 2, in the broadcast centre, the digital video signal is first compressed (or bit rate reduced), using the MPEG-2 compressor 2002. This compressed signal is then transmitted to the multiplexer and scrambler 2004 via the linkage 2006 in order to be multiplexed with other data, such as other compressed data.

[0045] The scrambler generates a control word CW used in the scrambling process and included in the MPEG-2 stream in the multiplexer 2004. The control word CW is generated internally and enables the end user's integrated receiver/decoder 2020 to descramble the programme. Access criteria, indicating how the programme is commercialised, are also added to the MPEG-2 stream. The programme may be commercialised in either one of a number of ((subscription)) modes and/or one of a number of ((Pay Per View)) (PPV) modes or events.

[0046] In the subscription mode, the end user subscribes to one or more commercial offers, of ((bouquets)), thus getting the rights to watch every channel inside those bouquets. In the preferred embodiment, up to 960 commercial offers may be selected from a bouquet of channels. In the Pay Per View mode, the end user is provided with the capability to purchase events as he wishes. This can be achieved by either pre-book- ing the event in advance ((pre-book mode)), or by purchasing the event as soon as it is broadcast ((impulse mode)).

[0047] Both the control word CW and the access criteria are used to build an Entitlement Control Message (ECM); this is a message sent in relation with a scrambled program. The message contains a control word (which allows for the descrambling of the program) and the access criteria of the broadcast program. The access criteria and control word are transmitted to the second encrypting unit 3014 via the linkage 3018. In this unit an ECM is generated, encrypted with an exploitation key Cex and transmitted on to the multiplexer and scrambler 2004.

Programme Transmission

[0048] The multiplexer 2004 receives encrypted EMMs from the SAS 3002, encrypted ECMs from the second encrypting unit 3014 and compressed programmes from the compressor 2002. The multiplexer 2004 scrambles the programmes and communicates the scrambled programmes, the encrypted EMM (if present) and the encrypted ECMs to a transmitter 2008 of the broadcast centre via linkage 2010. The transmitter 2008 transmits electromagnetic signals towards the satellite transponder 2014 via uplink 2012.

Programme Reception

[0049] The satellite transponder 2014 receives and processes the electromagnetic signals transmitted by the transmitter 2008 and transmits the signals on to the earth receiver 2018, conventionally in the form of a dish owned or rented by the end user, via downlink 2016. The signals received by receiver 2018 are transmitted to the integrated receiver/decoder 2020 owned or rented by the end user and connected to the end user's television set 2022. The receiver/decoder 2020 demultiplexes the signals to obtain scrambled programmes with encrypted EMMs and encrypted ECMs.

[0050] If the programme is not scrambled the receiver/decoder 2020 decompresses the data and transforms the signal into a video signal for transmission to television set 2022.

[0051] If the programme is scrambled, the receiver/decoder 2020 extracts the corresponding ECM from the MPEG-2 stream and passes the ECM to the ((daughter)) smart card 3020 of the end user. This slots into a housing in the receiver/decoder 2020. The daughter smart card 3020 controls whether the end user has the right to decrypt the ECM and to access the programme. If not, a negative status is passed to the receiver/decoder 2020 to indicate that the programme cannot be descrambled. If the end user does have the rights, the ECM is decrypted and the control word extracted. The decoder 2020 can then descramble the programme using this control word. The MPEG-2 stream is decompressed and translated into a video signal onward transmission to television set 2022.

Subscriber Management System (SMS)

[0052] A Subscriber Management System (SMS) 3004 includes a database 3024 which manages, amongst others, all of the end user files, commercial offers (such as tariffs and promotions), subscriptions, PPV details, and data regarding end user consumption and authorization. The SMS may be physically remote from the SAS

[0053] Each SMS 3004 transmits messages to the SAS 3002 via respective linkage 3006 to enable modifications to or creations of Entitlement Management Mes-

sages (EMMs) to be transmitted to end users.

[0054] The SMS 3004 also transmits messages to the SAS 3002 which imply no modifications or creations of EMMs but imply only a change in an end user's state (relating to the authorization granted to the end user when ordering products or to the amount that the end user will be charged).

Entitlement Management Messages and Entitlement Control Messages

[0055] ECMs or Entitlement Control Messages are encrypted messages embedded in the data stream of a transmitted program and which contain the control word necessary for descrambling of part or all of a program. Authorisation of a given receiver/decoder is controlled by EMMs or Entitlement Management Messages, transmitted on a less frequent basis and which supply an authorised receiver/decoder with the exploitation key necessary to decode the ECM.

[0056] An EMM is a message dedicated to an individual end user (subscriber), or a group of end users. A group may contain a given number of end users. This organisation as a group aims at optimising the bandwidth; that is, access to one group can permit the reaching of a great number of end users.

[0057] Various specific types of EMM may be used. Individual EMMs are dedicated to individual subscribers, and are typically used in the provision of Pay Per View services. So-called ((Group)) subscription EMMs are dedicated to groups, of say, 256 individual users, and are typically used in the administration of some subscription services. This EMM has a group identifier and a subscribers' group bitmap

[0058] For security reasons, the control word CW embedded in an encrypted ECM changes on average every 10 seconds or so. In contrast, the exploitation key Cex used by the receiver to decode the ECM is changed every month or so by means of an EMM. The exploitation key Cex is encrypted using a personalised key corresponding to the identity of the subscriber or group of subscribers recorded on the smart card. If the subscriber is one of those chosen to receive an updated exploitation key Cex, the card will decrypt the message using its personalised key to obtain that month's exploitation key Cex.

[0059] The operation of EMMs and ECMs will be well-known to one skilled in the art and will not be described here in any more detail.

Receiver/Decoder Structure

[0060] Referring to Figure 3, the elements of a receiver/decoder 2020 or set-top box for use in a digital broadcast system and adapted to be used in the present invention will now be described. As will be understood, the elements of this decoder are largely conventional and their implementation will be within the

capabilities of one skilled in the art.

[0061] As shown, the decoder 2020 is equipped with several interfaces for receiving and transmitting data, in particular an MPEG tuner and demultiplexer 2040 for receiving broadcast MPEG transmissions, a serial interface 2041, a parallel interface 2042, and a modem 2028 for sending and receiving data via the telephone network. In this embodiment, the decoder also includes a first and second smart card reader 2030 and 2031, the first reader 2030 for accepting a subscription smart card containing decryption keys associated with the system and the second reader 2031 for accepting bank and other cards. As will be described, the use of a two-slot decoder, adapted to read bank cards, is an important aspect in the implementation of the gaming system of Figure 4.

[0062] The decoder also includes a receiver 2043 for receiving infra-red control signals from the handset remote control 2044 and a Peritel output for sending audiovisual signals to a television 2022 connected to the decoder. In certain cases it may be desired that the infra-red signals transmitted from the handset 2044 to receiver 2043 are subject to a simple scrambling/descrambling process to ensure that no useful information may be obtained by any third party monitoring the transmission.

[0063] Such algorithms will not be described in any detail, but may comprise, for example a symmetric algorithmic key known to both handset 2044 and receiver/decoder 2020. This may be varied from time to time, for example, by means of a modulating random number chosen by the receiver/decoder 2020 and displayed by the television 2022, the user then programming the handset 2044 with this number to ensure that the handset scrambles entered data using an encryption algorithm key equivalent to that used the receiver/decoder to decrypt the received infra-red signals.

[0064] Processing of digital signals received via the interfaces and generation of digital output signals is handled by a central control unit 2045. The software architecture of the control unit within the decoder may correspond to that used in a known decoder and will not be described here in any detail. It may be based, for example, on a virtual machine interacting via an interface layer with a lower level operating system implemented in the hardware components of the decoder. In terms of the hardware architecture, the decoder will be equipped with a processor, memory elements such as ROM, RAM, FLASH memory etc. as in known decoders.

[0065] Applications processed by the control unit 2045 may be resident applications stored in the ROM or FLASH of the decoder or applications broadcast and downloaded via the MPEG interface 2 of the decoder. Applications can include program guide applications, games, interactive services, teleshopping applications, as well as initiating applications to enable the decoder

to be immediately operational upon start-up and applications for configuring the decoder. Applications are stored in memory locations in the decoder and represented as resource files comprising graphic object description files, unit files, variables block files, instruction sequence files, application files, data files etc.

[0066] Conventionally, applications downloaded into the decoder via the broadcast link are divided into modules, each module corresponding to one or more MPEG tables. Each MPEG table may be divided into a number of sections. For data transfer via the serial and parallel ports, modules are also split into tables and sections, the size of the section depending on the channel used.

[0067] In the case of broadcast transmission, modules are transported in the form of data packets within respective types of data stream, for example, the video data stream, the audio data stream, a text data stream. In accordance with MPEG standards each packet is preceded by a Packet Identifier (PID) of 13 bits, one PID for every packet transported in the MPEG stream. A programme map table (PMT) contains a list of the different streams and defines the content of each stream according to the respective PID. A PID may alert the device to the presence of applications in the data stream, the PID being identified by the PMT table.

Gaming System Architecture

[0068] Referring now to Figure 4, there will now be described the elements and functioning of a gaming system according to an embodiment of the present invention. The gaming system includes the elements of the digital television system described and shown in Figures 1 and 2, which have been assigned the same reference numerals. Some elements, such as the digital compressor 2002 shown in Figure 1, have been omitted in order to focus on those aspects of the system which are pertinent to the present invention.

[0069] As shown, the gaming system additionally comprises a source of audiovisual information 4001 regarding the event which will form the subject of betting etc within the system. In the present case, the event has been represented as a horse race, and the present system is indeed particular adapted to gaming activities centred around televised live action sporting events. However, as will be understood, the present system may equally used to permit gambling in relation to other events, such as casino-type games, as well as computer generated games, pre-recorded events etc.

[0070] The system further comprises a central gaming computer means in the form of a gaming system server 4002, together with associated operating terminal or terminals 4003, adapted to generate odds, calculate winnings etc in relation to the gaming event. The gaming server 4002 is adapted to communicate with a receiver/decoder 2020 via the intermediate communication server or servers 3022. The connection between the gaming server 4002 and communication server

3022 may be implemented by an X25 Transpac link or via a dedicated line. The network link for the server is indicated broadly at 4010.

[0071] As described above, the communication server 3022 communicates with the receiver/decoder 2020 by means of a telephone link using the in-built modem of the receiver/decoder.

[0072] The gaming server may be equally adapted to send information to the receiver/decoder 2020 via a satellite link, indicated broadly at 4011, by injection of information into the multiplexer 2004 for subsequent integration in the transmitted MPEG stream.

[0073] As will be understood, all communications from the receiver/decoder 2020 to the gaming server 4002 are via the receiver/decoder modem and communication server 3022. In the case of communications from the gaming server 4002 to the receiver/decoder 2020, the choice of communication channel and communication means (MPEG satellite transmission or communication server/modem connection) may depend on the nature of the information to be transmitted.

[0074] Typically, the satellite link 4011 will be used to send data or information that may be updated on a daily basis or which may be received by any number of receiver/decoders in the park (odds for tomorrow's races etc). In particular, the satellite link may be used to download the application that needs to be installed in the receiver/decoder to enable the receiver/decoder to function in the gaming system.

[0075] In contrast, the modem link 4010 may be preferred for data that changes on a minute-by-minute basis or that is specific to a particular user (results of last race, current state of the account of the user etc).

[0076] In addition to handling gaming activities resulting from bets placed via the receiver/decoder 2020, for example as programmed in using the remote control 2044, the gaming server 4002 may also be adapted to manage bets to be placed by other input means, for example as placed by a phone service or as received by a "Minitel" type system, as used in France and other countries.

[0077] The gaming system server 4002 is additionally connected to a bank server network 4003 comprising one or more bank servers 4005, 4006. The bank server network may correspond to an existing network used to handle electronic payment transactions. The level of security and encryption in the communications between each of the elements of the gaming system will be described in more detail below in relation to the operation of the system.

Gaming System Operation

[0078] As mentioned in the introduction of the present application, gaming systems used in interactive television systems proposed to date have tended to use relatively laborious methods for settling accounts between the viewer and the central gaming authority, requiring

the viewer either to pay by a conventional method (cheque, telephone credit transfer etc) or to physically purchase an "electronic purse" in the form of a smart card or key containing a number of pre-paid credits that may be gambled.

[0079] The present embodiment differs from such systems in proposing a system architecture that enables a viewer to pay by means of a credit or debit card inserted in the decoder and by entering data into the system by means of the hand-held remote control. As mentioned above, the provision of a decoder provided with two distinct card readers 2030, 2031 enables the decoder to simultaneously hold a subscription card containing the viewers access rights (eg to the gaming channel) as well as interacting with a credit/debit card inserted in the decoder.

[0080] In order to comply with regulations concerning the use of credit/debit cards in gambling transactions, two different types of transactions need to be distinguished: (i) opening or re-crediting an account managed by the gaming system server and (ii) gambling the sums in this account.

Opening an account

[0081] In the present case, the card reader 2031 functions in a similar manner to a standard card reader used in banking terminals and the like to read and write data on a smart card presented in the reader. As with all card readers used in the banking field, communication between the terminal (in this case the decoder) and external servers is prohibited during the time that the card is being accessed by the terminal, i.e. for the time that the memory zones on the card are "open".

[0082] In order to open and credit an account with the gaming system server, the following steps are carried out during a first phase:

a) Using the handheld remote control, and as guided by the application loaded in the receiver/decoder, the user selects the option "open an account" and enters the sum of money that he wishes to transfer to this account.

b) After having introduced his credit card into the card reader slot 2031, the viewer is invited to enter his personal PIN code. The user has a maximum of two opportunities to enter the code, after which the receiver/decoder will refuse to accept any further entries and the transaction will be abandoned.

Note that in the case of sensitive information communicated to the receiver/decoder by the handset (in particular the PIN code) the data entered by the user on the key pad of the handset may be scrambled before transmission between the handset and decoder so as to prevent interception of this information by any third party. See above.

c) Assuming the code is correct, the smart card downloads certain information in response to a request from the receiver/decoder, including details of the last transactions, to enable the decoder to verify that the sum of transactions during a certain period is within, for example, the transaction limit of the card holder for that period.

d) The receiver/decoder then passes to the smart card information regarding the current transaction including the amount of the transaction, the date and time of the transaction, the details of the bank account to be credited in the transaction and so on. (The details of the account to be credited can be obtained by the decoder prior to the interrogation of the card from the gaming system server or the intermediate communications system server).

e) In the conventional manner, the smart card then calculates a first numeric certificate using this information, which is communicated to the receiver/decoder. The receiver/decoder writes the present transaction in the card and a second numeric certificate is calculated and communicated to the receiver/decoder. The memory zones of the smart card are then closed off.

The generation of a pair of numeric certificates is a specific security measure associated with the use of a receiver/decoder as transaction terminal.

Once the above steps have been carried out, the system then moves to a second phase involving communication between the receiver/decoder 2020, the intermediate communication server 3022 and the bank server 4005.

f) Before transferring any information, the receiver/decoder 2020 verifies the identity of the communication server 3022 by means of a public/private key system (eg using the RSA algorithm). In particular, the receiver/decoder generates a random number, which is transmitted to the server for encryption by a private key and returned to the receiver/decoder, which checks the encrypted value using the equivalent public key.

A simple handshake signal may also be provided by the decoder 2020 to identify itself to the server 3022.

g) Assuming the identity of the communication server is verified, the receiver/decoder 2020 sends to the communication server 3022 the details of the transaction to be carried out, including the first and second numeric certificate generated by the smart card.

h) The communication server 3022 then sends the transaction details to the first bank server 4005, which verifies the account of the user, and author-

ises (or not) the transaction and sends an acknowledgement of the transaction to the communication server. The transfer of money between the user's account and that of the central gaming authority will then be handled within the bank network 4004.

i) Once the communication server 3022 has received acknowledgement of the acceptance of the monetary transfer, a message will be sent to the receiver/decoder 2020 of the completion of the transfer and the operation will proceed to the next phase.

Note that the same steps a) to i) as used in the first two phases will also be carried out in the event that the user wishes to increase the credit in an existing gaming account.

The next phase in the opening of a gaming account involves communication between the receiver/decoder 2020, the communication server 3022 (and the SAS and SMS servers 3002, 3004) and the gaming server 4002. The information communicated between these servers is largely non-sensitive and may be communicated in clear, with the exception of the code word chosen by the user to obtain access to his gaming account.

j) Using the information (name, address etc) on the user held in the SAS and SMS servers 3002, 3004, the communication server prepares a request for opening of an account with the gaming system server 4002. This information has been gathered in the SMS server during the original procedure carried out when the user originally subscribed to the television service. The user is thus spared the inconvenience of repeating all this information when subscribing to the gaming service.

Note that in the event that SMS database reveals, for example, that the subscriber is in debt with the television service, the communication server may abort the opening of an account with the gaming service. This extra verification step may be carried out earlier, for example, at step g).

k) In one embodiment, the communication server 3022 may send the subscriber information to the receiver/decoder 2020 where it is displayed on the television 2022 for verification by the user. Once verified, the information is sent to the gaming system server 4002 where a gambling account is created by the server 4002.

l) The account information (account number etc) is then sent from the gaming server 4002, via the communication server 3022, to the receiver/decoder 2022. The user is then invited to choose a suitable code word for the account which will be demanded by the system at every opening of a gaming session. As for the PIN number, the infra-

red signal containing this information and sent between the remote control and the decoder may be scrambled by the remote to avoid interception and descrambled by decoder.

m) The code word is then encrypted by a public key of a public/private key pair held in the receiver/decoder 2020 and sent to the communication server 3022, where it is decrypted by the corresponding private key. In this case, for example, the same RSA key pair as used for the verification of the communication server may be used.

n) The code word is then re-encrypted by the communication server 3022 and sent to the gaming system server 4002 where it is decrypted and assigned to the user's account. In this case, a symmetric key algorithm, such as DES, may be advantageously used, for example, to permit two-way encrypted communication between the communication server 3022 and gaming server 4002.

Gambling with an existing gaming account

[0083] Once the user has set up and credited a gaming account with the gaming server 4002, all future gambling transactions will be handled between the receiver/decoder 2020 and the gaming system server 4002. At the start of every gaming session, the system server 4002 will demand the user's assigned code word, which will be communicated between the receiver/decoder and the gaming server, via the communications server, as described above.

[0084] For simplicity, and in order to permit a relatively rapid dialogue, all questions and responses between the user and the gaming system in order to place a bet and receive the results are preferably passed via the telephone/modem link and the communication server 3022. Certain data, such as the format of the screens displayed by the receiver/decoder in gaming mode and/or slowly changing or universal data (details of that day's races, the horses taking part etc) may be passed via the satellite uplink in order to take advantage of the bandwidth of this channel.

[0085] Other embodiments, in which data is shared between the two communication channels in alternative ways may nevertheless be envisaged, for example, where all communication from the receiver/decoder to the gaming system server passes via the modem link, whilst all communications from the server to the receiver/decoder pass via the satellite link.

[0086] As mentioned above, the present system may be used with a number of interactive gaming applications, for example, with computer games such as blackjack, poker or the like, in which the user places a bet on the outcome of a game managed by the gaming server. However, in view of the use of television broadcast technology, the system is particularly adapted to permit

gaming in relation to live action sporting events, such as televised horse, dog or camel racing.

[0087] Figure 5 is a flow diagram of the steps involved in the placing of a bet in relation to one or more broadcast horse races. In the present case, the bet is to be placed in respect of the present day's races, i.e. in "real time", and the odds quoted for the horses may depend on the time at which the bet is taken. In alternative embodiments, bets may be placed the day or week before the race or races in question.

[0088] Firstly, at step 5000, the user enters his code word and opens a betting session. At steps 5001 and 5002, he chooses the racecourse he is interested in and one of the races running at that racecourse, respectively. Depending on which race is running, the user may be offered a number of different standard types of bet, from a simple bet to more complex bets, including main and side bets.

[0089] As will be appreciated, the bet types offered may be determined according to the wishes of the gaming authority and may be based on any of the usual types of bet offered for an event of this type.

[0090] At step 5003, the user chooses the type of bet he wishes to place. In the case of a simple bet on one horse, the next step will be step 5004 where the user chooses the formula of the bet, ie whether the horse will win or be placed in the first three or four positions. At step 5005, the user chooses the horse he wishes to bet on.

[0091] In the case of a complex bet, the user then chooses from a combination of win, place or win/place at step 5007 and from one of a number of types of bet (single, combined, reduced field, full field) at step 5007. The user may decide, for example to choose one horse to win and/or one horse to be placed in the top three or four. Other combinations may be made presented to reflect the choice of bet normally available. At step 5008 the user chooses the horses he wishes to bet on.

[0092] At step 5009 the user chooses his stake, i.e. the sum to be extracted from the money deposited in his gaming account. At step 5010 confirmation of the stake to be gambled is demanded. At this time, the system may also indicate the overall odds for the bet or bets placed and the sum of money to be won. Assuming that the user confirms the bet, the bet is registered at step 5011.

[0093] Following the results of the race, the gaming system server calculates the winnings or losses for the user. These will be subtracted or added automatically to his gaming account. The user may demand at any time the position of his account.

[0094] In the event that the user eventually wishes to close the account or to transfer some of his winnings to his bank account, a message to this end may be sent by the user from the receiver/decoder 2020 to the gaming system server 4002 (Figure 4). At that time, the server 4002 will communicate with the bank server 4006 to organise a credit transfer to the user's bank account.

Since the identity and bank details of the owner of the receiver/decoder are already known, the server will only transfer money from the gaming account of the user to the bank account originally used in the setting up of the gaming account.

[0095] It will be understood that the present invention has been described above purely by way of example, and modifications of detail can be made within the scope of the invention.

[0096] Each feature disclosed in the description, and (where appropriate) the claims and drawings may be provided independently or in any appropriate combination.

[0097] In the aforementioned preferred embodiments, certain features of the present invention have been implemented using computer software. However, it will of course be clear to the skilled man that any of these features may be implemented using hardware. Furthermore, it will be readily understood that the functions performed by the hardware, the computer software, and such like are performed on or using electrical and like signals.

Claims

1. An interactive gaming and audiovisual transmission system comprising a central gaming computer means for processing gaming data, a decoder adapted to receive gaming data from the central gaming computer together with transmitted audiovisual data, the decoder further including a card reading device for interacting with a user's bank card in order to credit a gaming account held by the central gaming computer means in response to a transfer of credit from the user's bank account.
2. An interactive gaming and audiovisual transmission system as claimed in claim 1, in which the decoder is equipped with a card reading device in the form of a smart card reader.
3. An interactive gaming and audiovisual transmission system as claimed in claim 1 or 2, in which the decoder is further equipped with a second card reading device
4. An interactive gaming and audiovisual transmission system as claimed in any preceding claim in which the decoder is adapted to obtain transfer of credit information in the form of an electronic certificate generated by the bank card in response to transaction data submitted by the decoder.
5. An interactive gaming and audiovisual transmission system as claimed in any preceding claim in which the decoder is provided with a handheld remote control, some or all of the data sent to the decoder being encrypted by the handheld remote

control and subsequently decrypted by the decoder.

6. An interactive gaming and audiovisual transmission system as claimed in any preceding claim in which the decoder is adapted to transmit transfer of credit information from the decoder to a bank server via a network communication link. 5

21. A gaming system as claimed in Claim 19 or 20, adapted to communicate with the decoder and the bank server via a communications server. 10

22. A gaming system as claimed in Claim 21, adapted to receive encrypted information from the communications server. 15

23. A gaming system as claimed in any of Claims 19 to 22, adapted to transmit gaming data related to a real-time sporting event. 20

24. An interactive gaming and audiovisual transmission system comprising a gaming system as claimed in any of Claims 19 to 23, said user's decoder, and said bank server. 25

30

35

40

45

50

55

Fig.1.

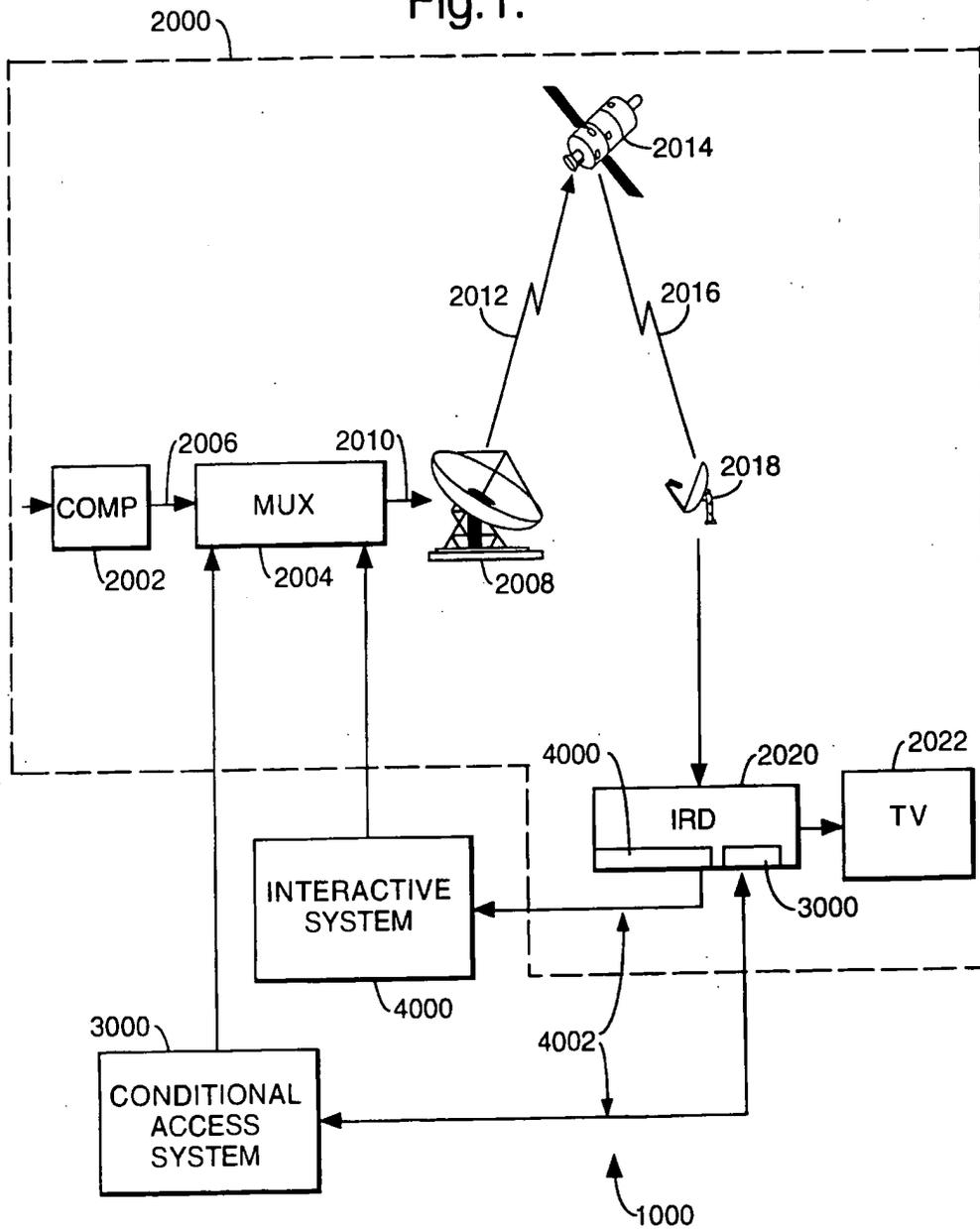


Fig.2.

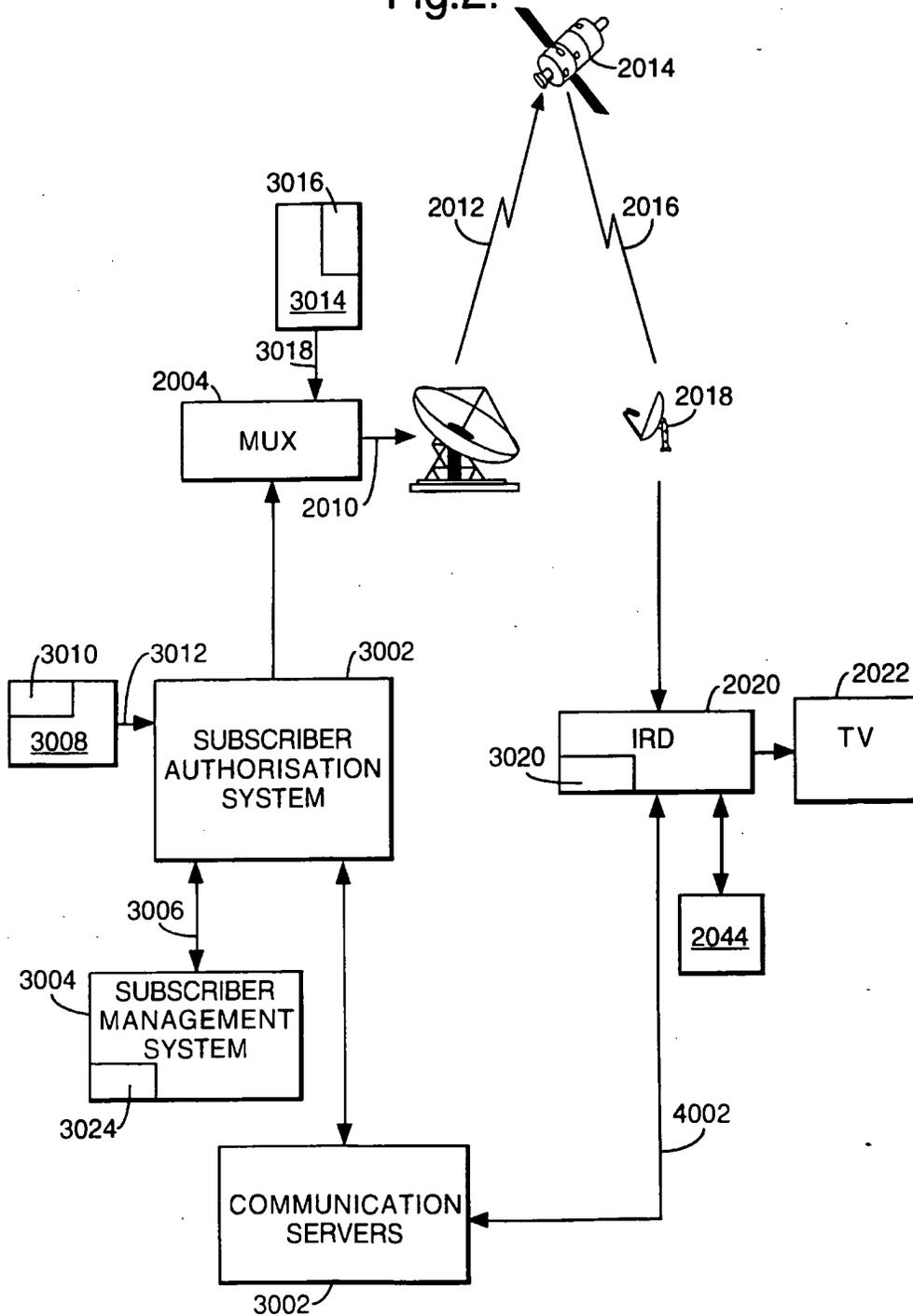
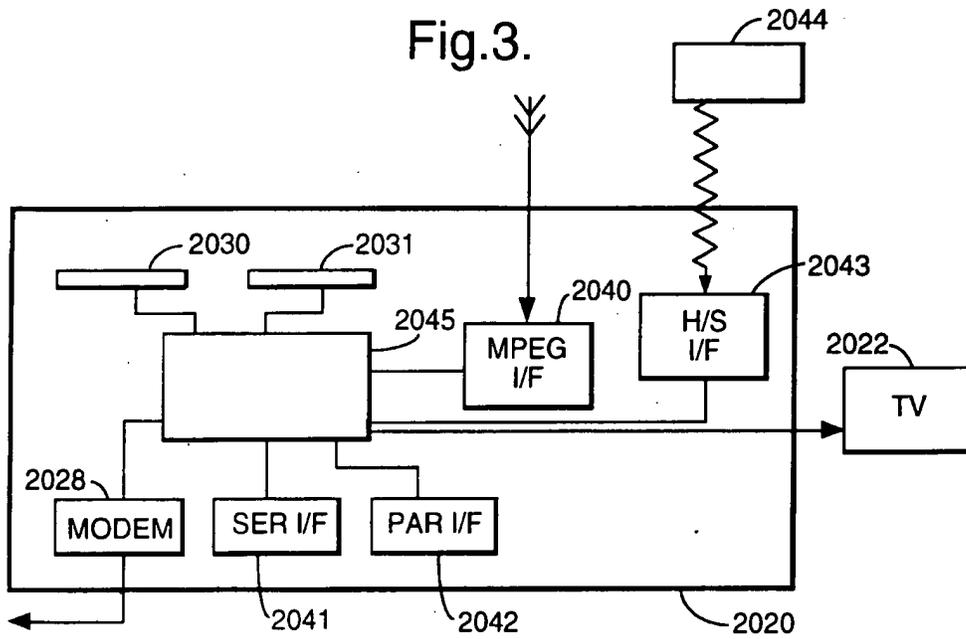


Fig.3.



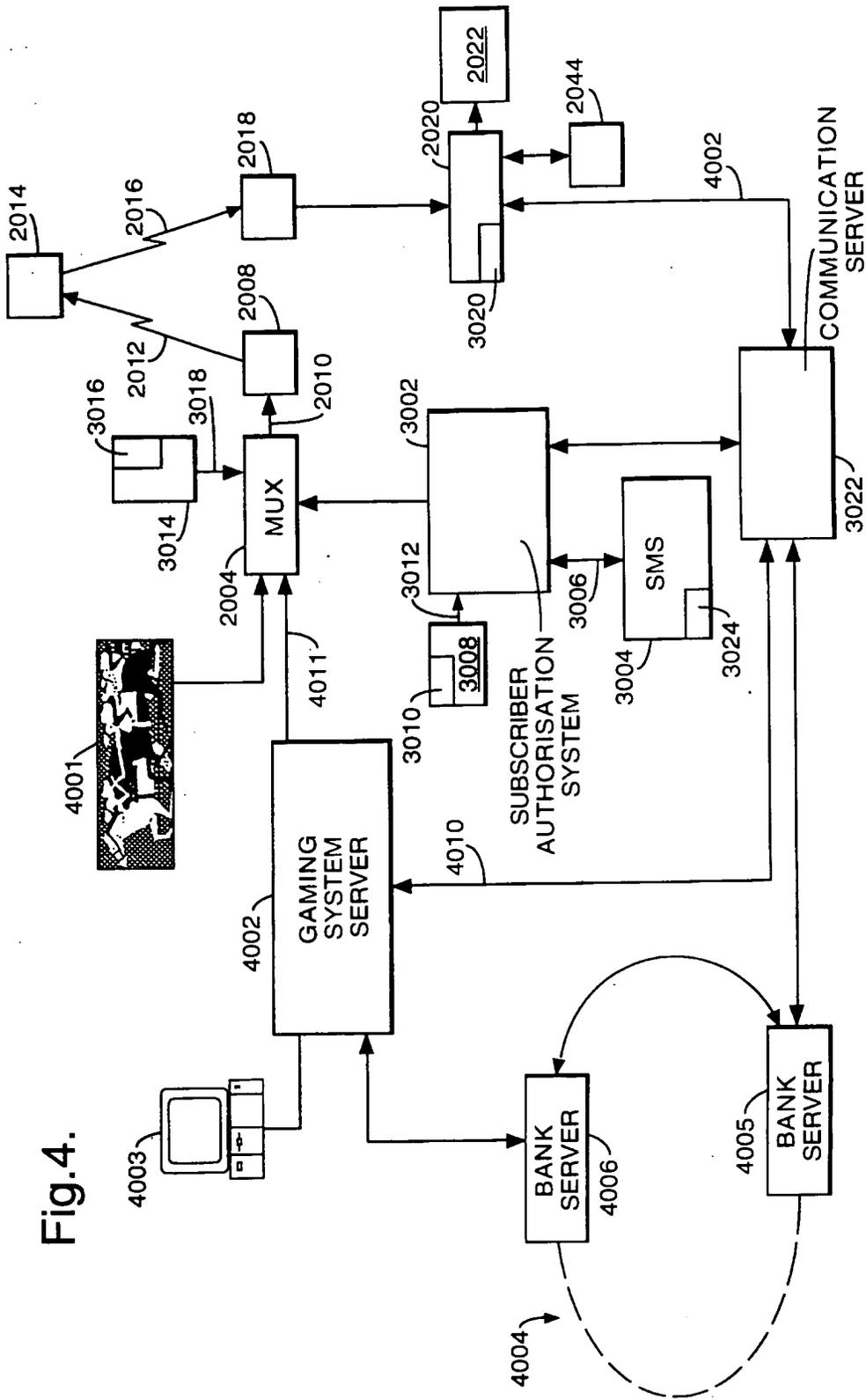


Fig.4.

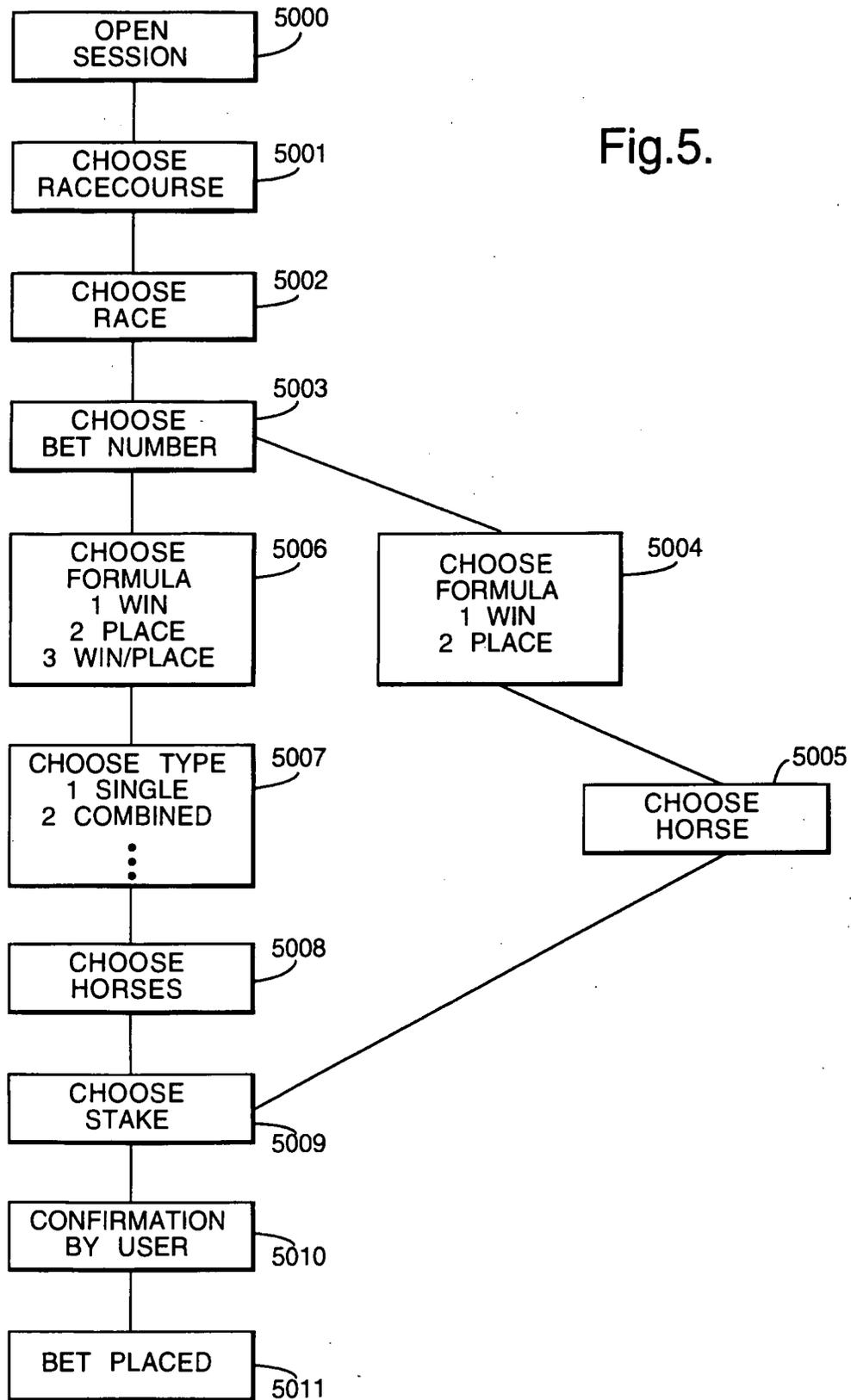


Fig.5.



European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 98 40 0285

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X Y A	US 5 539 822 A (LETT DAVID B) 23 July 1996 * column 7, line 13 - line 33 * * column 9, line 58 - column 10, line 11 * * column 11, line 22 - line 42 * * column 15, line 11 - line 44 * * column 18, line 44 - column 20, line 33 * * figures 3E, 3I * ---	1-3, 10, 12, 15-19, 23 4-9, 11, 14, 20, 21, 24 5	A63F9/22
X	US 4 815 741 A (SMALL MAYNARD E) 28 March 1989 * column 4, line 41 - column 5, line 5 * * column 5, line 27 - line 38 * ---	1, 19	
Y A	US 5 634 848 A (TSUDA YOICHIRO ET AL) 3 June 1997 * column 1, line 42 - line 64 * * column 3, line 6 - column 4, line 2 * * column 8, line 44 - column 9, line 21 * ---	6-9, 14, 20, 21, 24 1, 19	TECHNICAL FIELDS SEARCHED (Int.Cl.6) A63F H04N G07F G06F
Y	WO 95 01060 A (LINCOLN MINT HONG KONG LTD) 5 January 1995 * page 1, line 19 - line 29 * * page 3, line 17 - page 4, line 35 * * page 12, line 36 - page 13, line 35 * * page 14, line 31 - page 15, line 2 * * page 18, line 22 - line 27 * * page 20, line 8 - line 17 * * page 27, line 21 - page 28, line 6 * * page 29, line 4 - line 23 * * page 40, line 13 - page 42, line 2 * -----	4, 5, 11	
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 26 August 1998	Examiner Sindic, G
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

EPO FORM 1503 03/82 (P/4/C01)

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
 29.09.1999 Bulletin 1999/39

(51) Int. Cl.⁶: **H04L 12/58, H04L 29/06,**
H04L 12/22

(21) Application number: 99105140.0

(22) Date of filing: 26.03.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
 Designated Extension States:
AL LT LV MK RO SI

(72) Inventors:
 • Hisada, Yusuke
 Nippon Telegraph Telephone Corp
 Shinjuku-ku, Tokyo 163-14 (JP)
 • Ono, Satoshi
 Nippon Telegraph Telephone Corp
 Shinjuku-ku, Tokyo 163-14 (JP)
 • Ichikawa, Haruhisa
 Nippon Telegraph Telephone Corp
 Shinjuku-ku, Tokyo 163-14 (JP)

(30) Priority: 26.03.1998 JP 7983798
 18.06.1998 JP 17193098
 07.08.1998 JP 22486198
 05.11.1998 JP 31517298

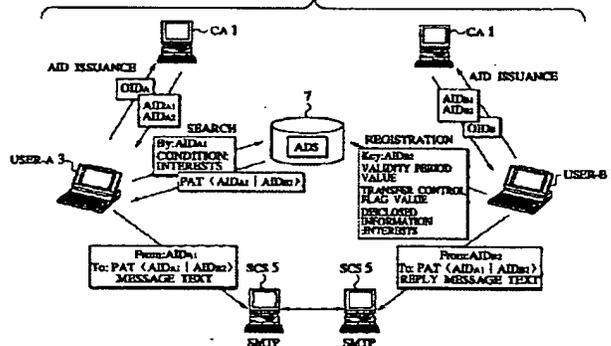
(74) Representative: **HOFFMANN - EITLÉ**
Patent- und Rechtsanwälte
Arabellastrasse 4
81925 München (DE)

(54) **Email access control scheme for communication network using identification concealment mechanism**

(57) An email access control scheme capable of resolving problems of the real email address and enabling a unique identification of the identity of the user while concealing the user identification is disclosed. A personalized access ticket containing a sender's identification and a recipient's identification in correspondence is to be presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email. Then, accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient

according to the personalized access ticket at a secure communication service. Also, an official identification of each user by which each user is uniquely identifiable by a certification authority, and an anonymous identification of each user containing at least one fragment of the official identification are defined, and each user is identified by the anonymous identification of each user in communications for emails on a communication network.

FIG.1



EP 0 946 022 A2

Description

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0001] The present invention relates to an email access control scheme for controlling transmission and reception of emails by controlling accesses for communications from other users whose identifications on the communication network are concealed while concealing an identification of a recipient on the communication network.

DESCRIPTION OF THE BACKGROUND ART

[0002] In conjunction with the spread of the Internet, the SPAM and the harassment using emails are drastically increasing. The SPAM is a generic name for emails or news that are unilaterally sent without any consideration to the recipient's time consumption, economical and mental burdens. The SPAM using emails are also known as UBE (Unsolicited Bulk Emails) or UCE (Unsolicited Commercial Emails).

[0003] The SPAM is sent indiscriminately regardless of the recipient's age, sex, interests, etc., so that the SPAM often contains an uninteresting or unpleasant content for the recipient. Moreover, the time consumption load and the economical load required for receiving the SPAM is not so small. For the business user, the SPAM can cause the lowering of the working efficiency as it becomes hard to find important mails that are buried among the SPAM. Also, as the SPAM is sent to a huge number of users, the SPAM wastes the network resources and in the worst case the SPAM can cause the overloading. As a result, there can be cases where mails that are important for the user may be lost. Also, the SPAM is sent either anonymously or by pretending someone else so that there is a need to provide some human resources to handle complaints.

[0004] On the other hand, the harassment is an act for keep sending mails with unpleasant contents for the user continually on the purpose of causing mental agony or exerting economical and time consumption burdens to the specific user. Similarly as the SPAM, the harassment mails are sent by pretending an actual or virtual third person, so that the identification of the sender is quite difficult. Also, there are cases where a large capacity mail is sent or a large amount of mails are sent in short period of time so that there is a danger of causing the system breakdown.

[0005] In order to deal with the SPAM and the harassment, the mail system is required to satisfy the following requirements.

Security

It is necessary to detect the pretending by the sender and refuse the delivery from the pretending

sender.

Strength

It is necessary to limit the mail capacity in order to circumvent the system breakdown due to the large capacity mail. It is also necessary to limit the number of transmissions in order to circumvent the system breakdown due to the large amount transmission.

Compatibility

It is necessary not to require a considerable change to the implementation of the existing mail system.

Handling

It is necessary not to require a considerable change to the handling of the existing mail system.

The MTA (message Transfer Agent) such as sendmail and qmail detects the forgery of the envelope information and the header information and refuses the delivery. The MTA also refuses mail receiving from a mail server which is a source of the SPAM by referring to the so called black list such as MAPS RBL. The MTA also detects the transmission using someone else's real email address and refuses the delivery by carrying out the signature verification using PGP, S/MIME, TLS, etc. The MTA also limits the message length by partial deletion of the message text.

One of the causes of the SPAM and the harassment is the real email address, and the real email address is associated with the following problems. User's identity can be guessed from real email address:

The real email address contains an information useful in guessing the identity so that it can be used in selecting the harassment target. For example, the place of employment can be identified from the real domain. Also, the name and the sex can be guessed from the user name.

Real email address can be guessed from user's identity:

The real email address has a universal format of [user name]@[domain name] so that the real email address can be guessed if the user's identity is known, without an explicit knowledge of the real email address itself. For example, if the user's real name is known, the candidates for the user name can be enumerated. Also, if the user's affiliation is known, the candidates for the domain name can be enumerated. Even in the case where the user name is given by a character string which is totally unrelated to the real name, if the naming rule for the user name is known, the user name can be guessed by trial and error transmissions.

Real email address is transferrable:

The real email address can be transferred from one person to another, so that mails can be transmitted even if the real email address is not taught by the holder himself. The transfer of real email

address through mails includes the following cases. By specifying the other's real email address in the cc: line of the mail, that real email address can be transferred to all the recipients specified in the To: line of the mail. Also, by forwarding the mail that contains the real email address of the recipient specified in the To: line in the message text to a third person, that real email address can be transferred to the third person.

Real email address is hard to cancel:

It is difficult to cancel the real email address because if the real email address is cancelled it becomes impossible to read not only the SPAM and the harassment mails but also the important mails as well.

[0006] Cypherpunk remailers and Mixmaster remailers which are collectively known as Anonymous remailers use a scheme for delivering mails after encrypting the real email address and the real domain of the sender. This scheme is called the reply block. The encryption and decryption of the reply block uses a public key and a secret key of the Anonymous remailer so that it is difficult to identify the real email address and the real domain of the sender for any users other than the sender.

[0007] The Anonymous remailers also make it difficult to transfer the real email address because it is difficult to identify the real email address. However, the reply block is transferrable, so that reply mails can be returned to the sender from users other than the recipient.

[0008] AS-Node and nym.alias.net which are collectively known as Pseudonymous servers use mail transmission and reception using a pseudonym account uniquely corresponding to the real email address of the user. The pseudonym account can be arbitrarily created at the user side so that the user can have a pseudonym account from which the real email address is hard to guess. In addition, by the use of the reply block, it is also possible to conceal the real email address and the real domain of the user to the Pseudonymous server. By combining these means, it can be made difficult to identify the real email address and the real domain of the sender for any users other than the sender. Also, the pseudonym account is cancellable so that there is no need to cancel the real email address.

[0009] The Pseudonymous servers also make it difficult to transfer the real email address because it is difficult to identify the real email address. However, the pseudonym account is transferrable so that reply mails can be returned to the sender from users other than the recipient.

[0010] In addition, in order to protect a recipient from the SPAM and the harassment, it is also necessary to reject a connection request from a sender who are exercising such action. For this reason, it is necessary for the communication system to be capable of uniquely identifying the identity of the sender.

[0011] In view of these factors, the communication system is required to be capable of uniquely identifying the identity of the user while concealing the real email address of the user (that is while guaranteeing the anonymity of the user), but in the conventional communication system, it has been difficult to meet both of these requirements simultaneously.

[0012] In order to identify the identity of the user in the mail system, the real email address of that user is necessary. On the other hand, the Anonymous remailers deliver a mail after either encrypting or deleting the real email address of the sender in order to guarantee the anonymity of the sender. In order to identify the identity of the sender under this condition, it is necessary to trace the delivery route of the mail using the traffic analysis. However, the Anonymous remailers may delay the mail delivery or interchange the delivery orders of mails. Also, The Mixmaster remailers deliver the mail by dividing it into plural blocks. For this reason, it is difficult to trace the delivery route by the traffic analysis, and therefore the identification of the identity of the sender is also difficult.

[0013] The Pseudonymous servers also utilize the Anonymous remailers for the mail delivery, so that it is possible to guarantee the anonymity of the sender but it is also difficult to uniquely identify the identity of the sender.

[0014] On the other hand, the German Digital Signature Law allows entry of a pseudonym instead of a real name into a digital certificate for generating the digital signature to be used in communication services. The digital certificate is uniquely assigned to the user so that the identity of the user can be uniquely identified even if the pseudonym is entered. Also, the right for naming the pseudonym is given to the user side so that it is possible to enter the pseudonym from which it is difficult to guess the real name.

SUMMARY OF THE INVENTION

[0015] It is therefore an object of the present invention to provide an email access control scheme in a communication network which is capable of resolving the above described problems of the real email address which is one of the causes of the SPAM and the harassment.

[0016] It is another object of the present invention to provide an email access control scheme in a communication network which is capable of enabling a unique identification of the identity of the user while concealing the user identification.

[0017] In order to resolve the problems associated with the transfer and the cancellation of the real email address, the present invention employs the email access control scheme using a personalized access ticket (PAT). In order to resolve the problem associated with the transfer of the real email address, the destination is specified by the PAT which contains both the real email address of the sender and a real email address of

the recipient. Also, in order to resolve the problem associated with the cancellation of the real email address, a validity period is set in the PAT by a Trusted Third Party. Then, the mail delivery from the sender who presented the PAT with the expired validity period will be refused. Also, instead of cancelling the real email address, the PAT is registered at a secure storage device managed by a secure communication service.

[0018] In other words, the present invention controls accesses in units in which the real email address of the sender and the real email address of the recipient is paired. For this reason, even when the real email address is transferred, it is possible to avoid receiving mails from users to which the real email address has been transferred as long as the PAT is not acquired by these users.

[0019] Also, in the present invention, it is possible to refuse receiving mails without cancelling the real email address because the mail delivery from the sender who presented the PAT with the expired validity period or the PAT that is registered in a database by the recipient will be refused.

[0020] Also, in the present invention, the mail receiving can be resumed without re-acquiring the real email address because the mail receiving can be resumed by deleting the PAT from the above described storage device.

[0021] Also, in the present invention, the time consumption and economical loads required for the mail receiving or downloading at the user side can be reduced because the transmission of mails are refused at the server side.

[0022] In addition, the present invention employs the email access control scheme using an official identification (OID) and an anonymous identification (AID) in order to make it possible to identify the identity of the user while guaranteeing the anonymity of the user.

[0023] Namely, in the present invention, a certificate in which the personal information is signed by a secret key of the Trusted Third Party is assigned to each user in order to uniquely identify each user. This certificate will be referred to as OID. Also, a certificate which contains fragments of the OID information is assigned to each user as a user identifier on a communication network in order to make it possible to identify the identity while guaranteeing the anonymity of the user. This certificate will be referred to as AID.

[0024] Also, in the present invention, the OID is reconstructed by judging the identity of a plurality of AIDs in order to identify the identity of the user. Also, the AID is contained in the PAT and the PAT is authenticated at a secure communication service (SCS) in order to resolve the problems associated with the transfer and the cancellation of the AID.

[0025] Also, in the present invention, the AID is managed in a directory which is accessible for search by unspecified many and which outputs the PAT containing the AID as a destination, in order to meet the user side

demand for being able to admit accesses from unspecified many without revealing the own identity.

[0026] In this way, in the present invention, the identity of the user can be concealed in the mail transmission and reception because the AID only contains fragments of the OID. Also, the identity of the user can be concealed from unspecified many even when the AID is registered at the directory service which is accessible from unspecified many.

[0027] Also, in the present invention, the identity of the user can be identified probabilistically by reconstructing the OID by judging the identity of a plurality of AIDs. For this reason, it is possible to provide a measure against the SPAM and the harassment without revealing the identity.

[0028] Also, in the present invention, it is possible to admit accesses from unspecified many without revealing the identity, by managing the AID rather than the real email address at the directory and outputting the PAT containing the AID as a destination at the directory.

[0029] More specifically, according to one aspect of the present invention there is provided a method of email access control, comprising the steps of: receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

[0030] Also, in this aspect of the present invention, at the controlling step the secure communication service authenticates the personalized access ticket presented by the sender, and refuses a delivery of the email when the personalized access ticket presented by the sender has been altered.

[0031] Also, in this aspect of the present invention, the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and at the controlling step the secure communication service authenticates the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

[0032] Also, in this aspect of the present invention, at the receiving step the secure communication service also receives the sender's identification presented by the sender along with the personalized access ticket, and at the controlling step the secure communication service checks whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuses a delivery of the email when the sender's identification presented by the sender is not contained in the person-

alized access ticket presented by the sender.

[0033] Also, in this aspect of the present invention, the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and at the controlling step the secure communication service checks the validity period contained in the personalized access ticket presented by the sender and refuses a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

[0034] Also, in this aspect of the present invention, the validity period of the personalized access ticket is set by a trusted third party.

[0035] Also, in this aspect of the present invention, the method can further comprise the step of: issuing the personalized access ticket to the sender at a directory service for managing an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

[0036] Also, in this aspect of the present invention, the method can further comprise the step of: registering in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service; wherein the controlling step the secure communication service refuses a delivery of the email from the sender when the personalized access ticket presented by the sender is registered therein in advance at the registering step.

[0037] Also, in this aspect of the present invention, the method can further comprise the step of: deleting the personalized access ticket registered at the secure communication service upon request from the specific registrant who registered the personalized access ticket at the registering step.

[0038] Also, in this aspect of the present invention, the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service, and at the controlling step, when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the secure communication service authenticates the sender's identification presented by the sender and refuses a delivery of the email when an authentication of the sender's identification fails.

[0039] Also, in this aspect of the present invention, the authentication of the sender's identification is realized

by a challenge/response procedure between the sender and the secure communication service.

[0040] Also, in this aspect of the present invention, the transfer control flag of the personalized access ticket is set by a trusted third party.

[0041] Also, in this aspect of the present invention, the sender's identification and the recipient's identification in the personalized access ticket can be given by real email addresses of the sender and the recipient.

[0042] Also, in this aspect of the present invention, the sender's identification and the recipient's identification in the personalized access ticket can be given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority.

[0043] Also, in this aspect of the present invention, the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority using a secret key of the certification authority.

[0044] Also, in this aspect of the present invention, the official identification of each user is a character string uniquely assigned to each user by the certification authority and a public key of each user which are signed by a secret key of the certification authority.

[0045] Also, in this aspect of the present invention, the method can further comprise the step of: probabilistically identifying an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0046] Also, in this aspect of the present invention, an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified can be defined, and the sender's identification and the recipient's identification in the personalized access ticket can be given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient.

[0047] Also, in this aspect of the present invention, the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority.

[0048] Also, in this aspect of the present invention, the method can further comprise the step of: probabilistically identifying an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

[0049] Also, in this aspect of the present invention, the personalized access ticket can contain a single sender's identification and a single recipient's identification in 1-to-1 correspondence.

[0050] Also, in this aspect of the present invention, the personalized access ticket can contain a single sender's identification and a plurality of recipient's identifications in 1-to-N correspondence, where N is an integer greater than 1.

[0051] Also, in this aspect of the present invention, one identification among the single sender's identification and the plurality of recipient's identifications is a holder identification for identifying a holder of the personalized access ticket while other identifications among the single sender's identification and the plurality of recipient's identifications are member identifications for identifying members of a group to which the holder belongs.

[0052] Also, in this aspect of the present invention, the method can further comprise the step of: issuing an identification of each user and an enabler of the identification of each user indicating a right to change the personalized access ticket containing the identification of each user as the holder identification, to each user at a certification authority, such that prescribed processing on the personalized access ticket can be carried out at a secure processing device only by a user who presented both the holder identification contained in the personalized access ticket and the enabler corresponding to the holder identification to the secure processing device.

[0053] Also, in this aspect of the present invention, the certification authority issues the enabler of the identification of each user as an information indicating that it is the enabler and the identification of each user itself which are signed by a secret key of the certification authority.

[0054] Also, in this aspect of the present invention, the prescribed processing includes a generation of a new personalized access ticket, a merging of a plurality of personalized access tickets, a splitting of one personalized access ticket into a plurality of personalized access tickets, a changing of the holder of the personalized access ticket, changing of a validity period of the personalized access ticket, and a changing of a transfer control flag of the personalized access ticket.

[0055] Also, in this aspect of the present invention, a special identification and a special enabler corresponding to the special identification which are known to all users can be defined such that the generation of a new personalized access ticket and the changing of the holder of the personalized access ticket can be carried out by the holder of the personalized access ticket by using the special identification and the special enabler without using an enabler of a member identification.

[0056] Also, in this aspect of the present invention, the special identification is defined to be capable of being used only as the holder identification of the personal-

ized access ticket.

[0057] Also, in this aspect of the present invention, a special identification which is known to all users can be defined such that a read only attribute can be set to the personalized access ticket by using the special identification.

[0058] Also, in this aspect of the present invention, at the controlling step, when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the secure communication service takes out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, converts the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and gives the mail after conversion to the mail transfer function by attaching the personalized access ticket.

[0059] According to another aspect of the present invention there is provided a method of email access control, comprising the steps of: defining an official identification of each user by which each user is uniquely identifiable by a certification authority, and an anonymous identification of each user containing at least one fragment of the official identification; and identifying each user by the anonymous identification of each user in communications for emails on a communication network.

[0060] Also, in this aspect of the present invention, the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority using a secret key of the certification authority.

[0061] Also, in this aspect of the present invention, the official identification of each user is a character string uniquely assigned to each user by the certification authority and a public key of each user which are signed by a secret key of the certification authority.

[0062] Also, in this aspect of the present invention, the method can further comprise the steps of: receiving a personalized access ticket containing a sender's anonymous identification and a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

[0063] Also, in this aspect of the present invention, the method can further comprises the step of: probabilistically identifying an identity of the sender at the secure communication service by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender con-

tained in a plurality of personalized access tickets used by the sender.

[0064] Also, in this aspect of the present invention, the defining step can also define a link information of each anonymous identification by which each anonymous identification can be uniquely identified, and each anonymous identification can also contain the link information of each anonymous identification.

[0065] Also, in this aspect of the present invention, the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority.

[0066] Also, in this aspect of the present invention, the method can further comprises the steps of: receiving a personalized access ticket containing a link information of a sender's anonymous identification and a link information of a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

[0067] Also, in this aspect of the present invention, the method can further comprises the step of: probabilistically identifying an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

[0068] According to another aspect of the present invention there is provided a communication system realizing email access control, comprising: a communication network to which a plurality of user terminals are connected; and a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

[0069] Also, in this aspect of the present invention, the secure communication service device authenticates the personalized access ticket presented by the sender, and refuses a delivery of the email when the personalized access ticket presented by the sender has been altered.

[0070] Also, in this aspect of the present invention, the system further comprises: a secure processing device

for issuing the personalized access ticket which is signed by a secret key of the secure processing device; wherein the secure communication service device authenticates the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

[0071] Also, in this aspect of the present invention, the secure communication service device also receives the sender's identification presented by the sender along with the personalized access ticket, checks whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuses a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

[0072] Also, in this aspect of the present invention, the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the secure communication service device checks the validity period contained in the personalized access ticket presented by the sender and refuses a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

[0073] Also, in this aspect of the present invention, the system further comprises: a trusted third party for setting the validity period of the personalized access ticket.

[0074] Also, in this aspect of the present invention, the system can further comprise: a directory service device for managing an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and issuing the personalized access ticket to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

[0075] Also, in this aspect of the present invention, the secure communication service device can register in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, and refuse a delivery of the email from the sender when the personalized access ticket presented by the sender is registered therein in advance.

[0076] Also, in this aspect of the present invention, the secure communication service device can delete the personalized access ticket registered therein upon request from the specific registrant who registered the personalized access ticket.

[0077] Also, in this aspect of the present invention, the

personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the secure communication service device authenticates the sender's identification presented by the sender and refuses a delivery of the email when an authentication of the sender's identification fails.

[0078] Also, in this aspect of the present invention, the authentication of the sender's identification is realized by a challenge/response procedure between the sender and the secure communication service device.

[0079] Also, in this aspect of the present invention, the system further comprises a trusted third party for setting the transfer control flag of the personalized access ticket.

[0080] Also, in this aspect of the present invention, the sender's identification and the recipient's identification in the personalized access ticket can be given by real email addresses of the sender and the recipient.

[0081] Also, in this aspect of the present invention, the system can further comprise: a certification authority device for issuing an anonymous identification of each user which contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by the certification authority device; wherein the sender's identification and the recipient's identification in the personalized access ticket can be given by anonymous identifications of the sender and the recipient.

[0082] Also, in this aspect of the present invention, the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority device using a secret key of the certification authority device.

[0083] Also, in this aspect of the present invention, the official identification of each user is a character string uniquely assigned to each user by the certification authority device and a public key of each user which are signed by a secret key of the certification authority device.

[0084] Also, in this aspect of the present invention, the secure communication service device can probabilistically identify an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0085] Also, in this aspect of the present invention, the system can further comprise: a certification authority device for issuing an anonymous identification of each user which contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by the certification authority device and a link information of each anonymous identification

by which each anonymous identification can be uniquely identified; wherein the sender's identification and the recipient's identification in the personalized access ticket can be given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient.

[0086] Also, in this aspect of the present invention, the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority device.

[0087] Also, in this aspect of the present invention, the secure communication service device can probabilistically identify an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

[0088] Also, in this aspect of the present invention, the personalized access ticket can contain a single sender's identification and a single recipient's identification in 1-to-1 correspondence.

[0089] Also, in this aspect of the present invention, the personalized access ticket can contain a single sender's identification and a plurality of recipient's identifications in 1-to-N correspondence, where N is an integer greater than 1.

[0090] Also, in this aspect of the present invention, one identification among the single sender's identification and the plurality of recipient's identifications is a holder identification for identifying a holder of the personalized access ticket while other identifications among the single sender's identification and the plurality of recipient's identifications are member identifications for identifying members of a group to which the holder belongs.

[0091] Also, in this aspect of the present invention, the system can further comprise: a certification authority device for issuing to each user an identification of each user and an enabler of the identification of each user indicating a right to change the personalized access ticket containing the identification of each user as the holder identification; and a secure processing device at which prescribed processing on the personalized access ticket can be carried out only by a user who presented both the holder identification contained in the personalized access ticket and the enabler corresponding to the holder identification to the secure processing device.

[0092] Also, in this aspect of the present invention, the certification authority device issues the enabler of the identification of each user as an information indicating that it is the enabler and the identification of each user itself which are signed by a secret key of the certification authority device.

[0093] Also, in this aspect of the present invention, the prescribed processing includes a generation of a new personalized access ticket, a merging of a plurality of

personalized access tickets, a splitting of one personalized access ticket into a plurality of personalized access tickets, a changing of the holder of the personalized access ticket, changing of a validity period of the personalized access ticket, and a changing of a transfer control flag of the personalized access ticket.

[0094] Also, in this aspect of the present invention, a special identification and a special enabler corresponding to the special identification which are known to all users can be defined such that the generation of a new personalized access ticket and the changing of the holder of the personalized access ticket can be carried out by the holder of the personalized access ticket by using the special identification and the special enabler without using an enabler of a member identification.

[0095] Also, in this aspect of the present invention, the special identification is defined to be capable of being used only as the holder identification of the personalized access ticket.

[0096] Also, in this aspect of the present invention, a special identification which is known to all users can be defined such that a read only attribute can be set to the personalized access ticket by using the special identification.

[0097] Also, in this aspect of the present invention, when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the secure communication service device takes out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, converts the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and gives the mail after conversion to the mail transfer function by attaching the personalized access ticket.

[0098] According to another aspect of the present invention there is provided a communication system realizing email access control, comprising: a certification authority device for defining an official identification of each user by which each user is uniquely identifiable by the certification authority device, and an anonymous identification of each user which contains at least one fragment of the official identification; and a communication network on which each user is identified by the anonymous identification of each user in communications for emails on the communication network.

[0099] Also, in this aspect of the present invention, the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority device using a secret key of the certification authority device.

[0100] Also, in this aspect of the present invention, the official identification of each user is a character string uniquely assigned to each user by the certification authority device and a public key of each user which are signed by a secret key of the certification authority

device.

[0101] Also, in this aspect of the present invention, the system can further comprises: a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a sender's anonymous identification and a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

[0102] Also, in this aspect of the present invention, the secure communication service device can probabilistically identify an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0103] Also, in this aspect of the present invention, the certification authority device can also define a link information of each anonymous identification by which each anonymous identification can be uniquely identified, and each anonymous identification can also contain the link information of each anonymous identification.

[0104] Also, in this aspect of the present invention, the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority device.

[0105] Also, in this aspect of the present invention, the system can further comprise: a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a link information of a sender's anonymous identification and a link information of a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

[0106] Also, in this aspect of the present invention, the secure communication service device can probabilistically identify an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of link informations of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0107] According to another aspect of the present invention there is provided a secure communication service device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to connect communications

between the sender and the receiver, by receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

[0108] Also, in this aspect of the present invention, the computer software causes the computer hardware to authenticate the personalized access ticket presented by the sender, and refuse a delivery of the email when the personalized access ticket presented by the sender has been altered.

[0109] Also, in this aspect of the present invention, the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and the computer software causes the computer hardware to authenticate the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

[0110] Also, in this aspect of the present invention, the computer software causes the computer hardware to also receive the sender's identification presented by the sender along with the personalized access ticket, check whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuse a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

[0111] Also, in this aspect of the present invention, the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the computer software causes the computer hardware to check the validity period contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

[0112] Also, in this aspect of the present invention, the computer software can cause the computer hardware to register in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service device, and refuse a delivery of the email from the sender when the personalized access ticket presented by the sender is registered at the secure communication service device in advance.

[0113] Also, in this aspect of the present invention, the computer software can cause the computer hardware to delete the personalized access ticket registered at the

secure communication service device upon request from the specific registrant who registered the personalized access ticket.

[0114] Also, in this aspect of the present invention, the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service device, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the computer software causes the computer hardware to authenticate the sender's identification presented by the sender and refuse a delivery of the email when an authentication of the sender's identification fails.

[0115] Also, in this aspect of the present invention, the computer software causes the computer hardware to realize the authentication of the sender's identification by a challenge/response procedure between the sender and the secure communication service device.

[0116] Also, in this aspect of the present invention, the sender's identification and the recipient's identification in the personalized access ticket can be given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority, and the computer software can also cause the computer hardware to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0117] Also, in this aspect of the present invention, an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified can be defined, the sender's identification and the recipient's identification in the personalized access ticket can be given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient, and the computer software can also cause the computer hardware to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

[0118] Also, in this aspect of the present invention, when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the computer software causes the computer hardware to take out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, convert

the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and give the mail after conversion to the mail transfer function by attaching the personalized access ticket.

[0119] According to another aspect of the present invention there is provided a secure processing device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to receive a request for a personalized access ticket from a user, and issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is signed by a secret key of the secure processing device.

[0120] According to another aspect of the present invention there is provided a directory service device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to manage an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

[0121] According to another aspect of the present invention there is provided a certification authority device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to issue to each user an official identification of each user by which each user is uniquely identifiable by the certification authority device, and an anonymous identification of each user which contains at least one fragment of the official identification.

[0122] According to another aspect of the present invention there is provided a certification authority device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to issue to each user an identification of each user and an enabler of the identification of each user indicating a right to change any personalized access ticket that contains the identification of each user as a holder identification, where the personalized access ticket generally contains a sender's identification and a plurality of recipient's identifications in correspondence, and one of the sender's identification and the recipient's identifications is a holder identification.

[0123] According to another aspect of the present

invention there is provided a secure processing device for use in a communication system realizing email access control, comprising: a computer hardware; and a computer software for causing the computer hardware to receive from a user a request for prescribed processing on a personalized access ticket containing a sender's identification and a plurality of recipient's identifications in correspondence, where one of the sender's identification and the recipient's identifications is a holder identification, and execute the prescribed processing on the personalized access ticket when the user presented both the holder identification contained in the personalized access ticket and an enabler corresponding to the holder identification which indicates a right to change the personalized access ticket containing the identification of the user as the holder identification.

[0124] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure communication service device for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to receive a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email; and second computer readable program code means for causing said computer to control accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket, so as to connect communications between the sender and the receiver on the communication network.

[0125] Also, in this aspect of the present invention, the second computer readable program code means causes said computer to authenticate the personalized access ticket presented by the sender, and refuse a delivery of the email when the personalized access ticket presented by the sender has been altered.

[0126] Also, in this aspect of the present invention, the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and the second computer readable program code means causes said computer to authenticate the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

[0127] Also, in this aspect of the present invention, the first computer readable program code means causes said computer to also receive the sender's identification presented by the sender along with the personalized access ticket, and the second computer readable program code means causes said computer to check

whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

[0128] Also, in this aspect of the present invention, the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the second computer readable program code means causes said computer to check the validity period contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

[0129] Also, in this aspect of the present invention, the second computer readable program code means can cause said computer to register in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service device, and refuse a delivery of the email from the sender when the personalized access ticket presented by the sender is registered at the secure communication service device in advance.

[0130] Also, in this aspect of the present invention, the second computer readable program code means can cause said computer to delete the personalized access ticket registered at the secure communication service device upon request from the specific registrant who registered the personalized access ticket.

[0131] Also, in this aspect of the present invention, the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service device, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the second computer readable program code means causes said computer to authenticate the sender's identification presented by the sender and refuse a delivery of the email when an authentication of the sender's identification fails.

[0132] Also, in this aspect of the present invention, the second computer readable program code means causes said computer to realize the authentication of the sender's identification by a challenge/response procedure between the sender and the secure communication service device.

[0133] Also, in this aspect of the present invention, the sender's identification and the recipient's identification in the personalized access ticket can be given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by

a certification authority, and the second computer readable program code means can also cause said computer to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

[0134] Also, in this aspect of the present invention, an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified can be defined, the sender's identification and the recipient's identification in the personalized access ticket can be given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient, and the second computer readable program code means can also cause said computer to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

[0135] Also, in this aspect of the present invention, when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the second computer readable program code means causes said computer to take out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, convert the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and give the mail after conversion to the mail transfer function by attaching the personalized access ticket.

[0136] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure processing device for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to receive a request for a personalized access ticket from a user; and second computer readable program code means for causing said computer to issue the personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is signed by a secret key of the secure processing device.

[0137] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as

a directory service device for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to manage an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and second computer readable program code means for causing said computer to issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

[0138] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a certification authority device for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to issue to each user an official identification of each user by which each user is uniquely identifiable by the certification authority device; and second computer readable program code means for causing said computer to issue to each user an anonymous identification of each user which contains at least one fragment of the official identification.

[0139] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a certification authority device for use in a communication system realizing email access control, the computer readable program code means includes: first computer readable program code means for causing said computer to issue to each user an identification of each user; and second computer readable program code means for causing said computer to issue to each user an enabler of the identification of each user indicating a right to change any personalized access ticket that contains the identification of each user as a holder identification, where the personalized access ticket generally contains a sender's identification and a plurality of recipient's identifications in correspondence, and one of the sender's identification and the recipient's identifications is a holder identification.

[0140] According to another aspect of the present invention there is provided a computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure processing device for use in a communication system realizing email access control, the computer

readable program code means includes: first computer readable program code means for causing said computer to receive from a user a request for prescribed processing on a personalized access ticket containing a sender's identification and a plurality of recipient's identifications in correspondence, where one of the sender's identification and the recipient's identifications is a holder identification; and second computer readable program code means for causing said computer to execute the prescribed processing on the personalized access ticket when the user presented both the holder identification contained in the personalized access ticket and an enabler corresponding to the holder identification which indicates a right to change the personalized access ticket containing the identification of the user as the holder identification.

[0141] Other features and advantages of the present invention will become apparent from the following description taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0142]

Fig. 1 is a diagram showing an overall configuration of a communication system according to the first embodiment of the present invention.

Fig. 2 is a diagram showing exemplary data structures of an official identification, an anonymous identification, and a 1-to-1 personalized access ticket according to the first embodiment of the present invention.

Fig. 3 is a flow chart for an anonymous identification generation processing at a certification authority according to the first embodiment of the present invention.

Fig. 4 is a flow chart for a personalized access ticket generation processing at an anonymous directory service according to the first embodiment of the present invention.

Fig. 5 is a flow chart for a mail access control processing at a secure communication service according to the first embodiment of the present invention.

Fig. 6 is a flow chart for an anonymous identification identity judgement processing at a secure communication service according to the first embodiment of the present invention.

Fig. 7 is a diagram showing exemplary data structures of data used in the anonymous identification identity judgement processing of Fig. 6.

Fig. 8 is a diagram showing exemplary data structures of an official identification, an anonymous identification, and a 1-to-N personalized access ticket according to the second embodiment of the present invention.

Fig. 9 is a diagram showing exemplary data struc-

tures of an anonymous identification and an enabler according to the second embodiment of the present invention.

Fig. 10 is a diagram showing a definition of a processing rule (MakePAT) used in the second embodiment of the present invention. 5

Fig. 11 is a diagram showing a definition of a processing rule (MergePAT) used in the second embodiment of the present invention.

Fig. 12 is a diagram showing a definition of a processing rule (SplitPAT) used in the second embodiment of the present invention. 10

Fig. 13 is a diagram showing a definition of a processing rule (TransPAT) used in the second embodiment of the present invention. 15

Fig. 14 is a first exemplary system configuration that can be used in the second embodiment of the present invention.

Fig. 15 is a second exemplary system configuration that can be used in the second embodiment of the present invention. 20

Fig. 16 is a third exemplary system configuration that can be used in the second embodiment of the present invention.

Fig. 17 is a fourth exemplary system configuration that can be used in the second embodiment of the present invention. 25

Fig. 18 is a fifth exemplary system configuration that can be used in the second embodiment of the present invention. 30

Fig. 19 is a sixth exemplary system configuration that can be used in the second embodiment of the present invention.

Fig. 20 is a seventh exemplary system configuration that can be used in the second embodiment of the present invention. 35

Fig. 21 is a flow chart showing an overall processing flow of MakePAT, MergePAT or TransPAT processing according to the second embodiment of the present invention. 40

Fig. 22 is a flow chart showing an overall processing flow of SplitPAT processing according to the second embodiment of the present invention.

Fig. 23 is a flow chart for an anonymous identification list generation processing (for MakePAT, MergePAT, SplitPAT and TransPAT) according to the second embodiment of the present invention. 45

Fig. 24 is an enabler authenticity verification processing (for MakePAT, MergePAT, SplitPAT and TransPAT) according to the second embodiment of the present invention. 50

Fig. 25 is a diagram showing an exemplary data structure of Null-AID used in the third embodiment of the present invention.

Fig. 26 is a diagram showing an exemplary data structure of Enabler of Null-AID used in the third embodiment of the present invention. 55

Fig. 27 is a diagram showing a first exemplary appli-

cation of the third embodiment of the present invention.

Fig. 28 is a diagram showing a second exemplary application of the third embodiment of the present invention.

Fig. 29 is a diagram showing an exemplary data structure of God-AID used in the fourth embodiment of the present invention.

Fig. 30 is a diagram showing a first exemplary application of the fourth embodiment of the present invention.

Fig. 31 is a diagram showing a second exemplary application of the fourth embodiment of the present invention.

Fig. 32 is a flow chart for a member anonymous identification checking processing according to the fifth embodiment of the present invention.

Fig. 33 is a diagram showing an overall configuration of a communication system according to the sixth embodiment of the present invention.

Fig. 34 is a diagram showing exemplary data structures of an official identification, a link information attached anonymous identification, and a link specifying 1-to-1 personalized access ticket according to the sixth embodiment of the present invention.

Fig. 35 is a flow chart for a link information attached anonymous identification generation processing at a certification authority according to the sixth embodiment of the present invention.

Fig. 36 is a flow chart for a link specifying 1-to-1 personalized access ticket generation processing at an anonymous directory service according to the sixth embodiment of the present invention.

Fig. 37 is a flow chart for a mail access control processing at a secure communication service according to the sixth embodiment of the present invention.

Fig. 38 is a flow chart for an anonymous identification identity judgement processing at a secure communication service according to the sixth embodiment of the present invention.

Fig. 39 is a diagram showing exemplary data structures of data used in the anonymous identification identity judgement processing of Fig. 38.

Fig. 40 is a diagram showing exemplary data structures of an official identification, a link information attached anonymous identification, and a link specifying 1-to-N personalized access ticket according to the seventh embodiment of the present invention.

Fig. 41 is a diagram showing exemplary data structures of a link information attached anonymous identification and an enabler according to the seventh embodiment of the present invention.

Fig. 42 is a first exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 43 is a second exemplary system configuration

that can be used in the seventh embodiment of the present invention.

Fig. 44 is a third exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 45 is a fourth exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 46 is a fifth exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 47 is a sixth exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 48 is a seventh exemplary system configuration that can be used in the seventh embodiment of the present invention.

Fig. 49 is a flow chart for a link specifying anonymous identification list generation processing (for MakePAT, MergePAT, SplitPAT and TransPAT) according to the seventh embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0143] Referring now to Fig. 1 to Fig. 7, the first embodiment of the email access control scheme according to the present invention will be described in detail.

[0144] The email access control scheme of the present invention enables bidirectional communications between a sender and a recipient appropriately while maintaining anonymity of a sender and a recipient on a communication network. Basically, this is realized by disclosing only information indicative of characteristics of recipients in a state of concealing true identifiers of the recipients, and assigning limited access rights with respect to those who wish to carry out communications while maintaining the anonymity according to the disclosed information.

[0145] More specifically, an Anonymous Identification (abbreviated hereafter as AID) that functions as a role identifier in which a personal information is concealed is assigned to a user, and this AID is disclosed on the network in combination with an information indicative of characteristics of the user such as his/her interests, age, job, etc., which cannot be used in identifying the user on the network but which can be useful for a sender in judging whether or not it is worth communicating with that user.

[0146] Also, the sender can search out a recipient with whom he/she wishes to communicate by reading or searching through the disclosed information. Namely, in the case where the sender wishes to communicate with a recipient while maintaining his/her own anonymity, the sender specifies the AID of that recipient and acquires a Personalized Access Ticket (abbreviated hereafter as

PAT). The PAT contains the AIDs of the sender and the recipient as well as information regarding a transfer control flag and a validity period. The transfer control flag is used in order to determine whether a Secure Communication Service (abbreviated hereafter as SCS) to be described below carries out the authentication with respect to the sender. Namely, when the transfer control flag is set ON, the SCS will carry out the authentication such as signature verification for example, with respect to the sender at a time of the connection request. On the other hand, when the transfer control flag is set OFF, the SCS will give the connection request to a physical communication network to which the SCS is connected, without carrying out the authentication. In other words, the transfer control is used in order to verify whether or not the AID is properly utilized by the user to whom it is allocated by a Certification Authority (abbreviated hereafter as CA).

[0147] In the communication network realizing the email access control scheme of the present invention, the assignment of AIDs with respect to users, the maintenance of information disclosed in combination with AIDs, the issuance of PATs, and the email access control based on PATs are realized by separate organizations. This is because it is more convenient to realize them by separate organizations from a perspective of maintaining the security of the entire network, since security levels to be maintained in relation to respective actions are different. Note however that the maintenance of the disclosed information and the issuance of PATs may be realized by the same organization.

[0148] Fig. 1 shows an overall configuration of a communication system in this first embodiment, which is directed to the email service on Internet or Intranet.

[0149] In Fig. 1, the CA (Certification Authority) 1 has a right to authenticate an Official Identification (abbreviated hereafter as OID) that identifies each individual and a right to issue AIDs, and functions to generate AIDs from OIDs and allocate AIDs to users 3.

[0150] The SCS (Secure Communication Service) 5 judges whether or not to admit a connection in response to a connection request by an email from a user 3, according to the PAT (Personalized Access Ticket) presented from a user 3. The SCS 5 also rejects a connection request by an email according to a request from a user 3. The SCS 5 also judges the identity of OIDs according to a request from a user 3.

[0151] An Anonymous Directory Service (abbreviated hereafter as ADS) 7 is a database for managing the AID, the transfer control flag value, the validity period value, and the disclosed information (such as interests, which can be regarded as requiring a lower secrecy compared with a personal information such as name, telephone number, and real email address) of each user 3. The ADS 7 has a function to generate the PAT from the AID of a user 3 who presented search conditions, the AID of a user 3 who has been registering the disclosed information that matches the search conditions

in the ADS 7, the transfer control flag value given from a user 3 or administrators of the ADS, and the validity period value given from a user 3 or administrators of the ADS, and then allocate the PAT to a user 3 who presented the search conditions.

[0152] First, a series of processing from generating the AID from the OID according to a request from a user until allocating the AID to that user will be described.

[0153] Fig. 2 shows exemplary formats of the OID, the AID, and the PAT. As shown in a part (a) of Fig. 2, the OID is an information comprising an arbitrary character string according to a rule by which the CA 1 can uniquely identify the user and a public key, which is signed by the CA 1 using a secret key of the CA 1.

[0154] Also, as shown in a part (b) of Fig. 2, the AID is an information comprising fragments of the OID and their position information, redundant character strings, and an SCS information given by an arbitrary character string (host name, real domain name, etc.) by which a host or a domain that is operating the SCS 5 can be uniquely identified on the network, which is signed by the CA 1 using the secret key of the CA 1.

[0155] Also, as shown in a part (c) of Fig. 2, the PAT is an information comprising the transfer control flag, AID_g, AID₁, and the validity period, which is signed by the ADS 7 using a secret key of the ADS 7. Here, the transfer control flag value is defined to take either 0 or 1. Also, the validity period is defined by any one or combination of the number of times for which the PAT is available, the absolute time (UTC) by which the PAT becomes unavailable, the absolute time (UTC) by which the PAT becomes available, and the relative time (lifetime) since the PAT becomes available until it becomes unavailable.

[0156] Note that, as will be explained in the subsequent embodiments described below, in addition to the 1-to-1 PAT which sets one sender and one recipient in correspondence as described above, the present invention can also use a 1-to-N PAT which sets one sender and N recipients, as well as a link specifying PAT which specifies the AID by a link information that is capable of specifying the AID instead of specifying the AID itself in the PAT. The link specifying PAT can be either a link specifying 1-to-1 PAT or a link specifying 1-to-N PAT depending on the correspondence relationship between the sender and the recipients as described above. Namely, the PAT of the present invention can be given in four types: 1-to-1 PAT, 1-to-N PAT, link specifying 1-to-1 PAT, and link specifying 1-to-N PAT.

[0157] Next, a procedure by which the user 3 requests the AID to the CA 1 will be described. The user 3 generates a pair of a secret key and a public key. Then, the user 3 and the CA 1 carries out the bidirectional authentication using the OID of the user 3 and the certificate of the CA 1, and the user 3 transmits the public key to the CA 1 by arbitrary means. Here, there can be cases where communications between the user 3 and the CA 1 are to be encrypted.

[0158] Next, a procedure by which the CA 1 issues the AID to the user 3 in response to a request for the AID as described above will be described. Upon receiving the public key from the user 3, the CA 1 generates the AID.

Then, the CA 1 transmits the AID to the user 3 by arbitrary means. Upon receiving the AID from the CA 1, the user 3 stores the received AID into its storage device. Here, there can be cases where communications between the user 3 and the CA 1 are to be encrypted.

[0159] Next, the AID generation processing at the CA will be described with reference to Fig. 3.

[0160] In the procedure of Fig. 3, the CA 1 generates an information of a length equal to the total length L of the OID, and sets this information as a tentative AID (step S911). Then, in order to carry out the partial copying of the OID, values of parameters p_i and l_i for specifying a copying region are determined using arbitrary means such as random number generation respectively (step S913). Here, L is equal to the total length L of the OID, and l_i is an arbitrarily defined value within a range in which a relationship of $0 \leq l_i \leq L$ holds. Then, an information in a range between a position p_i to a position $p_i + l_i$ from the top of the OID is copied to the same positions in the tentative AID (step S915). In other words, this OID fragment will be copied to a range between a position p_i and a position $p_i + l_i$ from the top of the tentative AID. Then, the values of p_i and l_i are written into a prescribed range in the tentative AID into which the OID has been partially copied, in a form encrypted by an arbitrary means (step S917). Then, an SCS information given by an arbitrary character string (host name, real domain, etc.) that can uniquely identify a host or a domain that is operating the SCS 5 on the network is written into a prescribed range in the tentative AID into which these values are written (step S919). Then, the tentative AID into which the above character string is written is signed using a secret key of the CA 1 (step S921).

[0161] Next, a procedure for registering the AID of a user-B 3 and the disclosed information into the ADS 7 will be described. First, the bidirectional authentication by arbitrary means using the AID of the user-B 3 and the certificate of the ADS 7 is carried out between the user-B 3 who is a registrant and the ADS 7. Then, the user-B 3 transmits the transfer control flag value, the validity period value, and the disclosed information such as interests to the ADS 7. Then, the ADS 7 stores the transfer control flag value, the validity period value, and the entire disclosed information in relation to the AID of the user-B 3 in its storage device. Here, there can be cases where communications between the user-B 3 who is the registrant and the ADS 7 are to be encrypted.

[0162] Next, a procedure by which a user-A 3 searches through the disclosed information that is registered in the ADS 7 will be described. First, the bidirectional authentication by arbitrary means using the AID of the user-A 3 and the certificate of the ADS 7 is carried out between the user-A 3 who is a searcher and the

ADS 7. Then, the user-A 3 transmits arbitrary search conditions to the ADS 7. Then, the ADS 7 presents all the received search conditions to its storage device, and extracts the AID of a registrant which satisfies these search conditions. Then, the ADS 7 generates the PAT from the AID of the user-A 3, the AID of the registrant who satisfied all the search conditions, the transfer control flag value, and the validity period value. Then, the ADS 7 transmits the generated PAT to the user-A 3. Here, there can be cases where communications between the user-A 3 who is a searcher and the ADS 7 are to be encrypted. Note that the 1-to-1 PAT is generated as a search result of the ADS 7.

[0163] Next, the 1-to-1 PAT generation processing at the ADS 7 will be described with reference to Fig. 4.

[0164] First, an information of a prescribed length is generated, and this information is set as a tentative PAT (step S1210). Then, the AID of the user-A 3 who is a searcher and the AID of the user-B 3 who is a registrant are copied into a prescribed region of the tentative PAT (step S1215). Then, the transfer control flag value and the validity period value are written into respective prescribed regions of the tentative PAT into which the AIDs are copied (step S1217). Then, the tentative PAT into which these values are written is signed using a secret key of the ADS 7 (step S1219).

[0165] Next, the transfer control using the 1-to-1 PAT will be described. The transfer control is a function for limiting accesses to a user who has a proper access right from a third person to whom the PAT has been transferred or who has eavesdropped the PAT (a user who originally does not have the access right).

[0166] The ADS 7 and the user-B 3 of the registrant AID can prohibit a connection to the user-B 3 from a third person who does not have the access right, by setting a certain value in to the transfer control flag of the PAT.

[0167] When the transfer control flag value is set to be 1, the sender's AID is authenticated between the SCS 5 and the sender according to an arbitrary challenge/response process, so that even if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will not be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0168] On the other hand, when the transfer control flag value is set to be 0, no challenge/response process will be carried out between the SCS 5 and the sender, so that if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will also be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0169] Next, the email access control method at the SCS 5 will be described with reference to Fig. 5.

[0170] The sender specifies "[sender's AID]@[real domain of SCS of sender]" in From: line, and "[PAT]@[real domain of SCS of sender]" in To: line.

[0171] The SCS 5 acquires a mail received by an MTA

(Message Transfer Agent) such as SMTP (Simple Mail Transfer Protocol), and executes the processing of Fig. 5 as follows.

(1) The signature of the PAT is verified using a public key of the ADS 7 (step S1413).

When the PAT is found to have been altered (step S1415 YES), the mail is discarded and the processing is terminated (step S1416).

When the PAT is found to have been not altered (step S1415 NO), the following processing (2) is executed.

(2) The search is carried out by presenting the sender's AID to the PAT (steps S1417, S1419, S1421).

When an AID that completely matches with the sender's AID is not contained in the PAT (step S1423 NO), the mail is discarded and the processing is terminated (step S1416).

When an AID that completely matches with the sender's AID is contained in the PAT (step S1423 YES), the following processing (3) is executed.

(3) The validity period value of the PAT is evaluated (steps S1425, S1427).

When the PAT is outside the validity period (step S1427 NO), the mail is discarded and the processing is terminated (step S1416).

When the PAT is within the validity period (step S1427 YES), the following processing (4) is executed.

(4) Whether or not to authenticate the sender is determined by referring to the transfer control flag value of the PAT (steps S1431, S1433).

When the value is 1 (step S1433 YES), the challenge/response authentication between the SCS 5 and the sender is carried out, and the signature of the sender is verified (step S1435). When the signature is valid, the recipient is specified and the PAT is attached (step S1437). When the signature is invalid, the mail is discarded and the processing is terminated (step S1416).

When the value is 0 (step S1433 NO), the recipient is specified and the PAT is attached without executing the challenge/response authentication (step S1437).

[0172] Next, an exemplary challenge/response authentication between the SCS 5 and the sender will be described.

[0173] First, the SCS 5 generates an arbitrary information such as a timestamp, for example, and transmits the generated information to the sender.

[0174] Then, the sender signs the received information using a secret key of the sender's AID and transmits it along with a public key of the sender's AID.

[0175] The SCS 5 then verifies the signature of the received information using the public key of the sender's AID. When the signature is valid, the recipient is speci-

fied and the PAT is attached. When the signature is invalid, the mail is discarded and the processing is terminated.

[0176] Next, a method for specifying the recipient at the SCS 5 will be described. First, the SCS 5 carries out the search by presenting the sender's AID to the PAT, so as to acquire all the AIDs which do not completely match the sender's AID. All these acquired AIDs will be defined as recipient's AIDs hereafter. Then, for every recipient's AID, the real domain of SCS of recipient is taken out from the recipient's AID. Then, the recipient is specified in a format of "[recipient's AID]@[real domain of SCS of recipient]". Finally, the SCS 5 changes the sender from a format of "[sender's AID]@[real domain of SCS of sender]" to a format of "sender's AID".

[0177] Next, a method for attaching the PAT at the SCS 5 will be described. The SCS 5 attaches the PAT to an arbitrary position in the mail. The SCS 5 gives the mail to the MTA after specifying the sender and the recipient and attaching the PAT.

[0178] Note that all the processings described above are the same in the case of the 1-to-N PAT.

[0179] Next, a method of receiving refusal with respect to the PAT at the SCS 5 will be described.

[0180] Receiving refusal setting: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user transmits a registration command, his/her own AID, and arbitrary PATs to the SCS 5. Then, the SCS 5 verifies the signature of the received AID. If the signature is invalid, the processing of the SCS 5 is terminated. If the signature is valid, the SCS 5 next verifies the signature of each received PAT using a public key of the ADS. Those PATs with the invalid signature are discarded by the SCS 5. When the signature is valid, the SCS 5 carries out the search by presenting the received AID to each PAT. For each of those PATs which contain the AID that completely matches with the received AID, the SCS 5 presents the registration command and the PAT to the storage device such that the PAT is registered into the storage device. Those PATs which do not contain the AID that completely matches with the received AID are discarded by the SCS 5 without storing them into the storage device. Here, there can be cases where communications between the user and the SCS 5 are to be encrypted.

[0181] Receiving refusal execution: The SCS 5 carries out the search by presenting the PAT to the storage device. When a PAT that completely matches the presented PAT is registered in the storage device, the mail is discarded. When a PAT that completely matches the present PAT is not registered in the storage device, the mail is not discarded.

[0182] Receiving refusal cancellation: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user presents his/her own AID to the SCS 5. Then, the SCS 5 verifies the signature of the received AID. If the signa-

ture is invalid, the processing of the SCS 5 is terminated. If the signature is valid, the SCS 5 next presents the presented AID as a search condition to the storage device and acquire all the PATs that contain the presented AID, and then presents all the acquired PATs to the user. Then, the user selects all the PATs for which the receiving refusal is to be cancelled by referring to all the PATs presented from the SCS 5, and transmits all the selected PATs along with a deletion command to the SCS 5. Upon receiving the deletion command and all the PATs for which the receiving refusal is to be cancelled, the SCS 5 presents the deletion command and all the PATs received from the user to the storage device, such that all the received PATs are deleted from the storage device.

[0183] Note that the method of receiving refusal with respect to the 1-to-N PAT at the SCS 5 is the same as the method of receiving refusal with respect to the 1-to-1 PAT described above.

[0184] Note also the the case of returning of a mail from the user-B to the user-A is the same as in the case of transmitting a mail from the user-A to the user-B.

[0185] Next, the judgement of identity will be described with reference to Fig. 6 and Fig. 7.

(1) An initial value of a variable OID_M is defined as a bit sequence with a length equal to the total length L of the OID and all values equal to "0". Also, an initial value of a variable OID_V is defined as a bit sequence with a length equal to the total length of the OID and all values equal to "0" (step S2511).

(2) One AID is selected from a set of processing target AIDs, and the following bit processing is carried out (step S2513).

(a) Values of variables AID_M and AID_V are determined according to the position information contained in the AID (step S2515). Here, AID_M is defined as a bit sequence with a length equal to the total length L of the OID and a value of a position at which the OID information is defined is "1" while a value of a position at which the OID information is not defined is "0" (see Fig. 7). Also, AID_V is defined as a bit sequence with a length equal to the total length L of the OID and a value of a position at which the OID information is defined is an actual value of the OID information while a value of a position at which the OID information is not defined is 0 (see Fig. 7).

(b) AND processing of OID_M and AID_M is carried out and its result is substituted into a variable OVR_M (step S2517).

(c) AND processing of OVR_M and AID_M as well as AND processing of OVR_M and OID_M are carried out and their results are compared (step S2519). When they coincide, OR processing of OID_M and AID_M is carried out

and its result is substituted into OID_M (step S2521), while OR processing of OID_V and AID_V is also carried out and its result is substituted into OID_M (step S2523). On the other hand, when they do not coincide, the processing proceeds to the step S2525.

(d) An AID to be processed next is selected from a set of processing target AIDs. When at least one another AID is contained in the set, the steps S2513 to S2523 are executed for that another AID. When no other AID is contained in the set, the processing proceeds to the step S2527.

(e) Values of OID_M and OID_V are outputted (step S2527).

[0186] The value of OID_M that is eventually obtained indicates all positions of the OID information that can be recovered from the set of processing target AIDs. Also, the value of OID_V that is eventually obtained indicates all the OID information that can be recovered from the set of processing target AID. In other words, by using the values of OID_M and OID_V , it is possible to obtain the OID albeit probabilistically when the value of OID_V is used as a search condition, and it is possible to quantitatively evaluate a precision of the above search by a ratio OID_M/L with respect to the total length L of the OID.

[0187] As described above, in this first embodiment, the CA 1 which is a Trusted Third Party with high secrecy and credibility generates the AID in which the personal information is concealed, from the OID that contains the highly secret personal information such as name, telephone number, real email address, etc., according to a user request, and issues the AID to the user. By identifying the user by this AID on the communication network as well as in various services provided on the communication network, it becomes possible to provide both the anonymity guarantee and the identity guarantee for the user. In other words, it becomes possible for the user to communicate with another user without revealing the own real name, telephone number, email address, etc., to that another user, and it also becomes possible to disclose the disclosed information to unspecified many through the ADS 7 as will be described below.

[0188] The user registers the disclosed information, that is an information which is supposed to have a low secrecy compared with the personal information at the ADS 7. In the case of searching the disclosed information and the registrant AID, the searcher presents the AID of the searcher and arbitrary search conditions to the ADS 7. The ADS 7 then extracts the registrant AID that satisfies these search conditions, and generates the PAT from the AID of the searcher and the AID of the registrant who satisfied the search conditions, the transfer control flag value, and the validity period value.

[0189] In this 1-to-1 PAT, the transfer control flag value and the validity period value are set as shown a part (c)

of Fig. 2, and by setting up this validity period in advance, it is possible to limit connections from the sender.

[0190] It is also possible to prohibit connections from a third person who does not have the access right, by using the transfer control flag value. Namely, when the transfer control flag value is set to be 1, the sender's AID is authenticated between the SCS 5 and the sender according to an arbitrary challenge/response process, so that even if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will not be able to make a connection to the registrant of the ADS 7 through the SCS 5. On the other hand, when the transfer control flag value is set to be 0, no challenge/response process will be carried out between the SCS 5 and the sender, so that if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will also be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0191] It is also possible to make a connection request to the communication network such that a call for which the recipient is specified by the 1-to-1 PAT will be received by the recipient's AID or the sender's AID defined within the PAT. In addition, it is also possible to refuse receiving calls with the 1-to-1 PAT selected by the recipient among calls which are specified by the 1-to-1 PAT. It is also possible to cancel the receiving refusal of the calls with the 1-to-1 PAT selected by the recipient. In addition, as a measure against the sender who repeats the personal attach using a plurality of sender's AIDs by taking an advantage of the anonymity, it is possible to judge the identity of the OID from these plurality of sender's AIDs and it is possible to extract that OID at some probability.

[0192] Next, with references to Fig. 8 to Fig. 24, the second embodiment of the email access control scheme according to the present invention will be described in detail.

[0193] In contrast to the first embodiment described above which is directed to the case where a sender and a recipient are set in 1-to-1 correspondence, this second embodiment is directed to the case where a sender and recipients are set in 1-to-N correspondence and a generation of a new PAT and a content change of the existing PAT can be made by the initiative of a user. Here, the sender is either a holder of the PAT or a member of the PAT. Similarly, the recipient is either a holder of the PAT or a member of the PAT.

[0194] In general, a membership of a group communication (mailing list, etc.) is changing dynamically so that it is necessary for a host of the group communication to manage information on a point of contact such as telephone number, email address, etc., of each member. In contrast, in the case where it is only possible to newly generate a 1-to-1 PAT as in the first embodiment, the management of a point of contact is difficult. For example, it is difficult to manage the group collectively, and

even if it is given to the others for the purpose of the transfer control, it does not function as an address of the group communication such as mailing list.

[0195] In this second embodiment, in order to resolve such a problem, it is made possible to carry out a generation of a new 1-to-N PAT and a content change or the existing 1-to-N PAT by the initiative of a user.

[0196] First, the definition of various identifications used in this second embodiment will be described with references to Fig. 8 and Fig. 9.

[0197] As shown in a part (a) of Fig. 8, the OID is an information comprising an arbitrary character string (telephone number, email address, etc.) according to a rule by which the CA 1 can uniquely identify the user and a public key, which is signed by the CA 1.

[0198] Also, as shown in a part (b) of Fig. 8, the AID is an information comprising fragments of the OID and their position information, redundant character strings, and an SCS information given by an arbitrary character string (host name, real domain name, etc.) by which a host or a domain that is operating the SCS 5 can be uniquely identified on the network, which is signed by the CA 1.

[0199] Also, as shown in a part (c) of Fig. 8, the 1-to-N PAT is an information comprising two or more AIDs, a holder index, the validity period, the transfer control flag, and a PAT processing device identifier, which is signed using a secret key of the PAT processing device.

[0200] Here, one of the AIDs is a holder AID of this PAT, where the change of the information contained in the PAT such as an addition of AID to the PAT, a deletion of AID from the PAT, a change of the validity period in the PAT, a change of the transfer control flag value in the PAT, etc., can be made by presenting the holder AID and a corresponding Enabler to the PAT processing device.

[0201] On the other hand, the AIDs other than the holder AID that are contained in the PAT are all member AIDs, where a change of the information contained in the PAT cannot be made even when the member AID and a corresponding Enabler are presented to the PAT processing device.

[0202] The holder index is a numerical data for identifying the holder AID, which is defined to take a value 1 when the holder AID is a top AID in the AID list formed from the holder AID and the member AIDs, a value 2 when the holder AID is a second AID from the top of the AID list, or a value n when the holder AID is an n-th AID from the top of the AID list.

[0203] The transfer control flag value is defined to take either 0 or 1 similarly as in the case of the 1-to-1 PAT.

[0204] The holder AID is defined to be an AID which is written at a position of the holder index value in the AID list. The member AIDs are defined to be all the AIDs other than the holder AID.

[0205] The validity period is defined by any one or combination of the number of times for which the PAT is available, the absolute time (UTC) by which the PAT

becomes unavailable, the absolute time (UTC) by which the PAT becomes available, and the relative time (lifetime) since the PAT becomes available until it becomes unavailable.

[0206] The identifier of a PAT processing device (or a PAT processing object on the network) is defined as a serial number of the PAT processing device (or a distinguished name of the PAT processing object on the network). The secret key of the PAT processing device (or the PAT processing object on the network) is defined to be uniquely corresponding to the identifier.

[0207] Also, in this second embodiment, an Enabler is introduced as an identifier corresponding to the AID. As shown in Fig. 9, the Enabler is an information comprising a character string uniquely indicating that it is an Enabler and an AID itself, which is signed by the CA 1.

[0208] Next, the operations for a generation of a new PAT and a content change of the existing PAT will be described. Here, the following operations are defined at a secure PAT processing device on the communication terminal or a PAT processing object on the CA or on a network which is properly requested from the CA (which will also be referred to as a PAT processing device hereafter).

1. Editing of AID list:

A list of AIDs (referred hereafter as an AID list) contained in the PAT is edited using AIDs and Enabler. Else, the AID list is newly generated.

2. Setting of the validity period and the transfer control flag:

The validity period value and the transfer control flag value contained in the PAT are changed using an AID and Enabler. Also, a new validity period value and a new transfer control flag value are set in the newly generated AID list.

[0209] A user who presented the holder AID and the Enabler corresponding to this holder AID to the PAT processing device can edit the list of AIDs contained in the PAT. In this case, the following processing rules are used.

(1) Generating a new PAT (MakePAT) (see Fig. 10):

The AID list (ALIST<holder AID | member AID₁, member AID₂, , member AID_n>) is newly generated, and the validity period value and the transfer control flag value are set with respect to the generated ALIST.

$$\text{AID}_A + \text{AID}_B + \text{Enabler of AID}_B + \text{Enabler of AID}_A$$

$$\rightarrow \text{ALIST}\langle \text{AID}_A | \text{AID}_B \rangle$$

$$\text{ALIST}\langle \text{AID}_A | \text{AID}_B \rangle + \text{Enabler of AID}_A$$

$$+ \text{validity period value}$$

+ transfer control flag value

→ PAT<AID_A | AID_B>

(2) Merging PATs (MergePAT) (see Fig. 11):

A plurality of ALISTS of the same holder AID are merged and the validity period value and the transfer control flag value are set with respect to the merged ALIST.

ALIST<AID_A | AID_{B1}, AID_{B2}, >

+ ALIST<AID_A | AID_{C1}, AID_{C2}, >

+ Enabler of AID_A

→ ALIST<AID_A | AID_{B1}, AID_{B2}, , AID_{C1}, AID_{C2}, >

ALIST<AID_A | AID_{B1}, AID_{B2}, , AID_{C1}, AID_{C2}, >

+ Enabler of AID_A + validity period value

+ transfer control flag value

→ PAT<AID_A | AID_{B1}, AID_{B2}, , AID_{C1}, AID_{C2}, >

(3) Splitting a PAT (SplitPAT) (see Fig. 12):

The ALIST is split into a plurality of ALISTS of the same holder AID, and the respective validity period value and transfer control flag value are set with respect to each one of the split ALISTS.

ALIST<AID_A | AID_{B1}, AID_{B2}, , AID_{C1}, AID_{C2}, >

+ Enabler of AID_A

→ ALIST<AID_A | AID_{B1}, AID_{B2}, >

+ ALIST<AID_A | AID_{C1}, AID_{C2}, >

ALIST<AID_A | AID_{C1}, AID_{C2}, >

+ Enabler of AID_A + validity period value

+ transfer control flag value

→ PAT<AID_A | AID_{C1}, AID_{C2}, >

(4) Changing a holder of a PAT (TransPAT) (see Fig. 13):

The holder AID of the ALIST is changed, and the validity period value and the transfer control flag value are set with respect to the changed ALIST.

ALIST<AID_A | AID_B> + ALIST<AID_A | AID_{C1}, AID_{C2}, >

+ Enabler of AID_A + Enabler of AID_B

→ ALIST<AID_B | AID_{C1}, AID_{C2}, >

ALIST<AID_B | AID_{C1}, AID_{C2}, >

+ Enabler of AID_B + validity period value

+ transfer control flag value

→ PAT<AID_B | AID_{C1}, AID_{C2}, >

[0210] In the operation for setting the validity period value, in order to permit the setting of the validity period value only to a user who holds both the holder AID and the corresponding Enabler, the following operation is defined.

PAT<AID_A | AID_B> + Enabler of AID_A

+ validity period value

→ PAT<AID_A | AID_B>

[0211] In the operation for setting the transfer control flag value, in order to permit the setting of the transfer control flag value only to a user who holds both the holder AID and the corresponding Enabler, the following operation is defined.

PAT<AID_A | AID_B> + Enabler of AID_A

+ transfer control flag value

→ PAT<AID_A | AID_B>

[0212] Next, with references to Fig. 14 to Fig. 20, the overall system configuration of this second embodiment will be described. In Fig. 14 to Fig. 20, the user-A who has AID_A allocated from the CA stores AID_A and Enabler of AID_A in a computer of the user-A, and the input/output devices such as floppy disk drive, CD-ROM drive, communication board, microphone, speaker, etc., are connected. Else, AID_A and Enabler of AID_A are stored in a communication terminal (telephone, cellular phone, etc.) which has a storage device and a data input/output function.

[0213] Similarly, the user-B who has AID_B allocated from the CA stores AID_B and Enabler of AID_B in a computer of the user-B, and the input/output devices such as floppy disk drive, CD-ROM drive, communication board, microphone, speaker, etc., are connected. Else, AID_B and Enabler of AID_B are stored in a communication terminal (telephone, cellular phone, etc.) which has

a storage device and a data input/output function.

[0214] In the following, a procedure by which the user-A generates PAT<AID_A | AID_B> will be described.

(1) The user-A acquires AID_B and Enabler of AID_B using any of the following means.

- * AID_B and Enabler of AID_B are registered at the ADS 7, and it is waited until the user-A acquires them as a search result (Fig. 14).
- * AID_B and Enabler of AID_B are directly transmitted to the user-A by the email, signaling, etc. (Figs. 15, 16).
- * AID_B and Enabler of AID_B are stored in a magnetic, optic, or electronic medium such as floppy disk, CD-ROM, MO, IC card, etc., and this medium is given to the user-A. Else, it is waited until the user acquires them by reading this medium (Figs. 17, 18).
- * AID_B and Enabler of AID_B are printed on a paper medium such as book, name card, etc., and this medium is given to the user-A. Else, it is waited until the user-A acquires them by reading this medium (Figs. 19, 20).

(2) The user-A who has acquired AID_B and Enabler of AID_B by any of the means described in the above (1) issues the MakePAT command to the PAT processing device. This procedure is common to Fig. 14 to Fig. 20, and defined as follows.

- (a) The user-A requests the issuance of the MakePAT command by setting AID_A, Enabler of AID_A, AID_B, Enabler of AID_B, the validity period value, and the transfer control flag value into the communication terminal of the user-A.
- (b) The communication terminal of the user-A generates the MakePAT command.
- (c) The communication terminal of the user-A transmits the generated MakePAT command to the PAT processing device by means such as the email, signaling, etc. (the issuance of the MakePAT command).
- (d) The PAT processing device generates PAT<AID_A | AID_B> by processing the received MakePAT command according to Fig. 21 and Fig. 23. More specifically, this is done as follows.

AID_A + AID_B + Enabler of AID_B + Enabler of AID_A

→ ALIST<AID_A | AID_B>

ALIST<AID_A | AID_B> + Enabler of AID_A

+ validity period value + transfer control flag value

→ PAT<AID_A | AID_B>

(e) The PAT processing device transmits the generated PAT<AID_A | AID_B> to the communication terminal of the user-A, or to the communication terminal of the user-B according to the need, by means such as the email, signaling, etc.

(f) The communication terminal of the user-A (or the user-B) stores the received PAT<AID_A | AID_B> in the storage device of the communication terminal of the user-A.

[0215] The merging of PATs (MergePAT, Fig. 21, Fig. 23), the splitting of a PAT (SplitPAT, Fig. 22, Fig. 23), and the changing of a holder of a PAT (TransPAT, Fig. 21, Fig. 23) are also carried out by the similar procedure.

[0216] Next, the procedure of MakePAT, MergePAT and TransPAT will be described with reference to Fig. 21.

- (1) The holder AID is specified (step S4411).
- (2) All the member AIDs are specified (step S4412).
- (3) The AID list is generated from the specified holder AID and all the specified member AIDs (step S4413). More specifically, the specified holder AID and all the specified member AIDs are concatenated using arbitrary means.
- (4) A tentative PAT is generated using arbitrary means, similarly as in the case of a tentative AID (step S4414).
- (5) The generated AID list is copied to a prescribed region of the generated tentative PAT (step S4415).
- (6) The holder index value is written into the tentative pat to which the AID list has been copied (step S4416).
- (7) The transfer control flag value is written into the tentative PAT into which the holder index value has been written (step S4417).
- (8) The validity period value is written into the tentative PAT into which the transfer control flag value has been written (step S4418).
- (9) The PAT processing device identifier is written into the tentative PAT into which the validity period value has been written (step S4419).
- (10) The tentative PAT into which the PAT processing device identifier has been written is signed using the secret key of the PAT processing device (step S4420).

[0217] Next, the procedure of SplitPAT will be described with reference to Fig. 22.

- (1) The holder AID is specified (step S4511).
- (2) All the AIDs to be the member AIDs of the PATs after the splitting are specified (step S4512).
- (3) The AID list is generated from the specified holder AID and all the specified member AIDs (step

S4513). More specifically, the specified holder AID and all the specified member AIDs are concatenated using arbitrary means.

(4) A tentative PAT is generated using arbitrary means, similarly as in the case of a tentative AID (step S4514).

(5) The generated AID list is copied to a prescribed region of the generated tentative PAT (step S4515).

(6) The holder index value is written into the tentative pat to which the AID list has been copied (step S4516).

(7) The transfer control flag value is written into the tentative PAT into which the holder index value has been written (step S4517).

(8) The validity period value is written into the tentative PAT into which the transfer control flag value has been written (step S4518).

(9) The PAT processing device identifier is written into the tentative PAT into which the validity period value has been written (step S4519).

(10) The tentative PAT into which the PAT processing device identifier has been written is signed using the secret key of the PAT processing device (step S4520).

(11) In the case of continuing the splitting (step S4521 YES), the procedure returns to (2), and repeats (2) to (10) sequentially.

[0218] Note that, in the procedures of Fig. 21 and Fig. 22, the AID list generation is carried out according to Fig. 23 as follows. Namely, a buffer length is determined first (step S4611) and a buffer is generated (step S4612). Then, the holder AID is copied to a vacant region of the generated buffer (step S4613). Then, the member AID is copied to a vacant region of the resulting buffer (step S4614), and if the next member AID exists (step S4615 YES), the step S4614 is repeated.

[0219] Next, the determination of the holder AID will be described. Each of the MakePAT, the MergePAT, the SplitPAT, and the TransPAT commands is defined to have two or more arguments, where AID, PAT, or Enabler can be specified as an argument. In this case, the PAT processing device specifies the holder AID of the PAT to be outputted after executing each command according to the following rules.

Case of the MakePAT:

For the MakePAT command, it is defined that AIDs are to be specified for the first argument to the N-th argument (N = 2, 3,) and Enablers are to be specified for the N+1-th and subsequent arguments. For example, they can be specified as follows.

MakePAT AID₁, AID₂,, AID_N,
Enabler of AID₁, Enabler of AID₂, Enabler of AID_N

The PAT processing device interprets the AID of the first argument of the MakePAT command as the holder AID.

Only when one of the Enablers of the N+1-th and subsequent arguments corresponds to the AID of the first argument, the PAT processing device specifies this AID (that is the AID of the first argument) as the holder AID of the PAT to be outputted after executing the MakePAT command.

Case of the MergePAT:

For the MergePAT command, it is defined that PATs are to be specified for the first argument to the N-th argument (N = 2, 3,) and Enabler is to be specified for the N+1-th argument. Namely, they can be specified as follows.

MergePAT PAT₁ PAT₂ PAT_N Enabler of AID

The PAT processing device interprets the holder AID of the PAT of the first argument of the MergePAT command as the holder AID of the PAT to be outputted after executing the MergePAT command.

Only when the Enabler of the N+1-th argument corresponds to the holder AID of the PAT of the first argument, the PAT processing device specifies this AID (that is the holder AID of the PAT of the first argument) as the holder AID of the PAT to be outputted after executing the MergePAT command.

Case of the SplitPAT:

For the SplitPAT command, it is defined that PAT is to be specified for the first argument, a set of one or more AIDs grouped together by some prescribed symbols (assumed to be parentheses ()) in this example) are to be specified for the second argument to the N-th argument (N = 3, 4,), and Enabler is to be specified for the N+1-th argument. Namely, they can be specified as follows.

SplitPAT PAT₁ (AID₁₁) (AID₂₁ AID₂₂)
. (AID_{N1} AID_{N2}
AID_{NM}) Enabler of AID

The PAT processing device interprets the holder AID of the PAT of the first argument of the SplitPAT command as the holder AID of the PAT to be outputted after executing the SplitPAT command.

Only when the Enabler of the N+1-th argument corresponds to the holder AID of the PAT of the first argument, the PAT processing device specifies this AID (that is the holder AID of the PAT of the first argument) as the holder AID of the PAT to be outputted after executing the SplitPAT command.

Case of the TransPAT:

For the TransPAT command, it is defined that

PATs are to be specified for the first argument and the second argument, AID is to be specified for the third argument, and Enablers are to be specified for the fourth argument and the fifth argument. Namely, they can be specified as follows.

TransPAT PAT₁ PAT₂ AID Enabler of AID₁ Enabler of AID₂

The PAT processing device interprets the AID of the third argument as the holder AID of the PAT to be outputted after executing the TransPAT command provided that the AID of the third argument of the TransPAT command is contained in the PAT of the second argument.

Only when the Enabler of the fourth argument corresponds to both the PAT of the first argument and the PAT of the second argument and the Enabler of the fifth argument corresponds to the AID of the third argument, the PAT processing device specifies the AID of the third argument as the holder AID of the PAT to be outputted after executing the TransPAT command.

Next, the determination of the member AIDs will be described. The definitions of the MakePAT, the MergePAT, the SplitPAT, and the TransPAT commands are as described above. The PAT processing device specifies the member AIDs of the PAT to be outputted after executing each command according to the following rules.

Case of the MakePAT:

Only when the holder AID of the PAT to be outputted after executing the MakePAT command is formally determined, the PAT processing device interprets all the AIDs of the second and subsequent arguments of the MakePAT command as the member AIDs of the PAT to be outputted after executing the MakePAT command.

The PAT processing device specifies only those AIDs among all the AIDs of the second and subsequent arguments which correspond to the Enablers specified by the N+1-th and subsequent arguments as the member AIDs of the PAT to be outputted after executing the MakePAT command.

Case of the MergePAT:

Only when the holder AID of the PAT to be outputted after executing the MergePAT command is formally determined, the PAT processing device specifies the member AIDs of all the PATs specified by the first to N-th arguments of the MergePAT as the member AIDs of the PAT to be outputted after executing the MergePAT command.

Case of the SplitPAT:

Only when the holder AID of the PAT to be outputted after executing the SplitPAT command is formally determined, the PAT processing device specifies the member AID of the PAT specified by the first argument of the SplitPAT command as the

member AID of the PAT to be outputted after executing the SplitPAT command. At this point, the member AIDs are distributed into different PATs in units of parentheses (). For example, in the case of:

SplitPAT PAT (AID₁₁) (AID₂₁ AID₂₂)
..... (AID_{N1} AID_{N2}
AID_{NM}) Enabler of AID

(AID₁₁), (AID₂₁ AID₂₂) and (AID_{N1} AID_{N2} AID_{NM}) will be the member AIDs of different PATs having a common holder AID.

Case of TransPAT:

Only when the holder AID of the PAT to be outputted after executing the TransPAT command is formally determined, the PAT processing device specifies all the member AIDs remaining after excluding the member AID that is scheduled to be a new holder AID from all the member AIDs of the PAT specified by the first argument of the TransPAT command and the member AIDs of the PAT specified by the second argument as the member AIDs of the PAT to be outputted after executing the TransPAT command.

[0220] Next, the verification of the properness of the Enabler will be described. This verification of the properness of the Enabler is common to the MakePAT, the MergePAT, the SplitPAT and the TransPAT, and carried out according to Fig. 24 as follows.

- (1) AID and Enabler are entered (step S5511).
- (2) Each of these entered AID and Enabler is verified using the public key of the CA 1 (step S5512). If at least one of them is altered (step S5513 YES), the processing is terminated.
- (3) A character string for certifying that it is Enabler is entered (step S5514).
- (4) The top field of the Enabler of the step S5511 and the character string of the step S5514 are compared (step S5515). If they do not match (step S5516 NO), the processing is terminated.
- (5) If they match (step S5516 YES), the AID of the step S5511 and the AID within the Enabler are compared (step S5517).
- (6) A comparison result is outputted (step S5519).

[0221] Next, with references to Fig. 25 to Fig. 28, the third embodiment of the email access control scheme according to the present invention will be described in detail.

[0222] In the generation of a new PAT (MakePAT) and the PAT holder change (TransPAT) of the above described embodiment, it is necessary to give member AIDs and Enablers of member AIDs to the holder of the PAT, but when they are given to the holder, it becomes possible for that holder to participate the group communications hosted by the other holders by using the

acquired member AIDs. Namely, there arises a problem that the pretending using the member AIDs become possible. Moreover, if that holder places the acquired member AIDs and Enablers of member AIDs on a medium that is readable by unspecified many, these member AIDs become accessible to anyone so that there arises a problem that the harassment to the users of the member AIDs may occur and the pretending using the member AIDs by a third person also become possible.

[0223] For this reason, in this third embodiment, it is made possible to carry out the MakePAT and the TransPAT without giving the Enablers of member AIDs to the holder.

[0224] To this end, in this third embodiment, the generation of a new PAT and the content change of the existing PAT are carried out by using Null-AID (AID_{Null}) and Enabler of Null-AID (Enabler of AID_{Null}).

[0225] Here, the processing involving the Null-AID obeys all of the following rules:

- (a) the processing rules of MakePAT, MergePAT, SplitPAT and TransPAT as in the above described embodiment; and
- (b) the rules applicable only to the Null-AID, including:

- (i) Null-AID is known to every user, and
- (ii) Enabler of Null-AID is known to every user.

[0226] Here, the processing rules as defined in the above described embodiment in the case of this third embodiment will be described.

(1) Making a PAT from plural AIDs (MakePAT):

AID_{holder} + AID_{member1} + AID_{member2} +
 + AID_{memberN}
 + Enabler of AID_{member1} + Enabler of
 AID_{member2} +
 + Enabler of AID_{memberN} + Enabler of AID_{holder}
 → PAT<AID_{holder} | AID_{member1}, AID_{member2},
 , AID_{memberN} >

(2) Merging plural PATs of the same holder (MergePAT):

PAT<AID_{holder} | AID_{membera1}, AID_{membera2},
 , AID_{memberaM} >
 + PAT<AID_{holder} | AID_{memberb1}, AID_{memberb2},
 , AID_{memberbN} >
 + Enabler of AID_{holder}

→ PAT<AID_{holder} | AID_{membera1}, AID_{membera2},
 , AID_{memberaM}, AID_{memberb1},
 AID_{memberb2}, , AID_{memberbN} >

(3) Splitting a PAT into plural PATs of the same holder (SplitPAT):

PAT<AID_{holder} | AID_{membera1}, AID_{membera2},
 , AID_{memberaM}, AID_{memberb1},
 AID_{memberb2}, , AID_{memberbN} >
 + Enabler of AID_{holder}
 → PAT<AID_{holder} | AID_{membera1}, AID_{membera2},
 , AID_{memberaM} >
 + PAT<AID_{holder} | AID_{memberb1}, AID_{memberb2},
 , AID_{memberbN} >

(4) Changing a holder AID of a PAT (TransPAT):

PAT<AID_{holder} | AID_{membera1}, AID_{membera2},
 , AID_{memberaM} > + PAT<AID_{holder}
 | AID_{newholder} >
 + Enabler of AID_{holder} + Enabler of AID_{newholder}
 → PAT<AID_{newholder} | AID_{membera1},
 AID_{membera2}, , AID_{memberaM} >

[0227] The method for specifying the validity period value and the transfer control flag value in the PAT containing the Null-AID is similar to the method for specifying the validity period value and the transfer control flag value in the second embodiment described above. Next, the exemplary processings involving the Null-AID will be described.

(1) Case of producing PAT<AID_{Null} | AID_A > from AID_A and Enabler of AID_A:

- (a) According to the above described rules (b)(i) and (b)(ii) of the Null-AID, AID_{Null} and Enabler of AID_{Null} are known.
- (b) Using MakePAT,

AID_{Null} + AID_A + Enabler of AID_A + Enabler
 of AID_{Null}
 → PAT<AID_{Null} | AID_A >

(2) Case of producing PAT<AID_{Null} | AID_A, AID_B > from PAT<AID_{Null} | AID_A > and PAT<AID_{Null} | AID_B >:

- (a) According to the above described rules (b)(i) and (b)(ii) of the Null-AID, AID_{Null} and Enabler of AID_{Null} are known.

(b) Using MergePAT,

PAT<AID_{Null} | AID_A > + PAT<AID_{Null} | AID_B

>

+ Enabler of AID_{Null}

→ PAT<AID_{Null} | AID_A, AID_B >.

(3) Case of producing PAT<AID_A | AID_B > from PAT<AID_{Null} | AID_A >, PAT<AID_{Null} | AID_B > and Enabler of AID_A:

(a) According to the above described rules (b)(i) and (b)(ii) of the Null-AID, AID_{Null} and Enabler of AID_{Null} are known.

(b) Using TransPAT,

PAT<AID_{Null} | AID_A > + PAT<AID_{Null} | AID_B

>

+ Enabler of AID_{Null} + Enabler of AID_A

→ PAT<AID_A | AID_B >.

[0228] As shown in Fig. 25, the data structure of the Null-AID comprises a character string uniquely indicating that it is Null-AID (a character string defined by the CA, for example), which is signed by the CA using the secret key of the CA.

[0229] Also, as shown in Fig. 26, the data structure of the Enabler of Null-AID comprises a character string uniquely indicating that it is Enabler (a character string defined by the CA, for example) and the Null-AID itself, which is signed by the CA using the secret key of the CA.

[0230] Note that the Null-AID and the Enabler of Null-AID are maintained at secure PAT processing devices and secure PAT certification authority.

[0231] Next, the first exemplary application of this third embodiment will be described with reference to Fig. 27, which includes the following operations.

(1) The user-B (PAT member) generates PAT<AID_{Null} | AID_B > by executing the above described exemplary processing (1) involving the Null-AID at the secure PAT processing device which is connected with the terminal of the user-B, and gives it to the user-A (PAT holder) by arbitrary means.

(2) The user-A who received PAT<AID_{Null} | AID_B > carries out the following operations at the secure PAT processing device which is connected with the terminal of the user-A.

(a) PAT<AID_{Null} | AID_A > is produced by executing the above described exemplary processing (1) involving the Null-AID.

(b) PAT<AID_A | AID_B > is produced by execut-

ing the above described exemplary processing (3) involving the Null-AID.

(3) The user-A gives the generated PAT<AID_A | AID_B > to the user-B by arbitrary means.

[0232] Note that the method for determining the validity period is the same as described above so that it will not be repeated here. Also, the processing involving the Null-AID is the same as described above so that it will not be repeated here.

[0233] In the case of giving PAT<AID_{Null} | AID_A, AID_B > to the user-B, the above described exemplary processing (2) involving the Null-AID will be executed in the operation (2) described above.

[0234] Next, the second exemplary application of this third embodiment will be described with reference to Fig. 28, which includes the following operations.

(1) The user-B (PAT member) produces PAT<AID_{Null} | AID_B > by executing the above described exemplary processing (1) involving the Null-AID at the secure PAT processing device which is connected with the terminal of the user-B, and registers it along arbitrary disclosed information at the ADS.

(2) The user-A produces PAT<AID_{Null} | AID_A > by executing the above described exemplary processing (1) involving the Null-AID at the secure PAT processing device which is connected with the terminal of the user-A, and presents it along arbitrary search conditions to the ADS.

(3) When the personal information of the user-B satisfies the search conditions presented by the user-A, the secure PAT processing device connected with the ADS carries out the following operations.

(a) PAT<AID_{Null} | AID_A, AID_B > is produced by executing the above described exemplary processing (2) involving the Null-AID.

(b) The produced PAT<AID_{Null} | AID_A, AID_B > is given to the ADS.

(4) The ADS gives PAT<AID_{Null} | AID_A, AID_B > produced by the PAT processing device to the user-A.

(5) The user-A who received PAT<AID_{Null} | AID_A, AID_B > produces PAT<AID_A | AID_B > by executing the following TransPAT processing at the secure PAT processing device which is connected with the terminal of the user-A.

PAT<AID_{Null} | AID_A > + PAT<AID_{Null} | AID_A, AID_B >

+ Enabler of AID_{Null} + Enabler of AID_A

→ PAT<AID_A | AID_B >.

[0235] Note that the method for determining the validity period is the same as described above so that it will not be repeated here. Also, the processing involving the Null-AID is the same as described above so that it will not be repeated here.

[0236] In the case of generating $PAT\langle AID_A | AID_B \rangle$ at the PAT processing device connected with the ADS, Enabler of AID_A will be given to that PAT processing device, and the above described exemplary processing (3) involving the Null-AID will be executed in the operation (3) described above.

[0237] In the case of generating $PAT\langle AID_B | AID_A \rangle$ at the PAT processing device connected with the ADS and giving it to the user-B, Enabler of AID_B will be given to that PAT processing device, and the above described exemplary processing (3) involving the Null-AID will be executed in the operation (3) described above.

[0238] Next, with references to Fig. 29 to Fig. 31, the fourth embodiment of the email access control scheme according to the present invention will be described in detail.

[0239] In the group communication, a situation where it is desired to fix the participants is frequently encountered, but the above described embodiment does not have a function for making it impossible to change the PAT so that the participants cannot be fixed. Namely, in the above described embodiment, whether or not to fix the participants is left to the judgement of the holder of the PAT.

[0240] For this reason, in this fourth embodiment, a read only attribute is set up in the PAT. More specifically, in this fourth embodiment, the read only attribute is set up in the PAT by using God-AID (AID_{God}).

[0241] Here, the processing involving the God-AID obeys all of the following rules:

- (a) God-AID is known to every user, and
- (b) the processing involving God-AID is allowed only in the following cases:

(i) a case where the AID_{holder} is neither AID_{Null} nor AID_{God} :

$PAT\langle AID_{holder} | AID_{member1}, AID_{member2}, \dots, AID_{memberN} \rangle + \text{Enabler of } AID_{holder}$

$\rightarrow PAT\langle AID_{god} | AID_{holder}, AID_{member1}, AID_{member2}, \dots, AID_{memberN} \rangle$

(ii) a case where AID_{holder} is AID_{Null} :

$PAT\langle AID_{Null} | AID_{member1}, AID_{member2}, \dots, AID_{memberN} \rangle$

+ Enabler of AID_{Null}

$\rightarrow PAT\langle AID_{god} | AID_{member1}, AID_{member2}, \dots, AID_{memberN} \rangle$

....., $AID_{memberN} \rangle$

[0242] As shown in Fig. 29, the data structure of the God-AID comprises a character string uniquely indicating that it is God-AID (a character string defined by the CA, for example), which is signed by the CA using the secret key of the CA. The God-AID is maintained at the secure PAT processing devices and the secure PAT certification authority described above.

[0243] The processings of a PAT that contains the Null-AID are according to Fig. 21 to Fig. 24. When the holder AID is neither Null-AID nor God-AID, the God-AID is appended to the AID list and the holder index value is specified to be a position of the God-AID in the AID list after appending the God-AID. When the holder AID is Null-AID, the Null-AID is deleted from the AID list, the God-AID is appended to the AID list, and then the holder index value is specified to be a position of the God-AID in the AID list after appending the God-AID.

[0244] Next, the exemplary application of this fourth embodiment will be described with reference to Fig. 30.

[0245] In the case of producing $PAT\langle AID_{God} | AID_A, AID_B \rangle$ from $PAT\langle AID_{Null} | AID_A \rangle$ and $PAT\langle AID_{Null} | AID_B \rangle$, the following processing is executed at the secure PAT processing device which is connected with the terminal of the PAT holder (user-A in Fig. 30).

(1) Using MergePAT,

$PAT\langle AID_{Null} | AID_A \rangle + PAT\langle AID_{Null} | AID_B \rangle$

+ Enabler of AID_{Null}

$\rightarrow PAT\langle AID_{Null} | AID_A, AID_B \rangle$

(2) According to the above described rule (a) of the God-AID, AID_{God} is known.

(3) According to the above described rule (b)(i) of the God-AID,

$PAT\langle AID_{Null} | AID_A, AID_B \rangle + \text{Enabler of } AID_{Null}$

$\rightarrow PAT\langle AID_{god} | AID_A, AID_B \rangle$

[0246] The above processing is also executed at the secure PAT processing device connected with a computer (search engine, etc.) of the third person (Fig. 31) or at the secure PAT certification authority.

[0247] Next, with reference to Fig. 32, the fifth embodiment of the email access control scheme according to the present invention will be described in detail.

[0248] When the Null-AID is added as described in the third embodiment, there arises a problem that it becomes possible for the holder of the PAT (the user of the holder AID) to transfer the access right with respect to the member (the user of the member AID) to the third person, and moreover this transfer can be done without a permission of the member, as will be described now.

(1) The holder-A of PAT<AID_A | AID_B> (for the member-B) produces PAT<AID_{Null} | AID_B> by using PAT<AID_A | AID_B>, AID_A and Enabler of AID_A. Here, it is assumed that the holder-A knows all of AID_A, Enabler of AID_A, AID_{Null}, and Enabler of AID_{Null} in addition to PAT<AID_A | AID_B>.

(a) The holder-A produces PAT<AID_A | AID_{Null}> using the MakePAT as follows.

AID_A + AID_{Null} + Enabler of AID_{Null} + Enabler of AID_A

→ PAT<AID_A | AID_{Null}>

(b) The holder-A produces PAT<AID_{Null} | AID_B> using the TransPAT as follows.

PAT<AID_A | AID_B> + PAT<AID_A | AID_{Null}>

+ Enabler of AID_A + Enabler of AID_{Null}

→ PAT<AID_{Null} | AID_B>

After the above described operation (1)(b), the holder-A gives PAT<AID_{Null} | AID_B> to the third person-C, the following operation (2) becomes possible.

(2) The third person-C produces PAT<AID_C | AID_B> by using PAT<AID_{Null} | AID_B>. Here, it is assumed that the third person-C knows all of AID_C, Enabler of AID_C, AID_{Null}, and Enabler of AID_{Null} in addition to PAT<AID_{Null} | AID_B>.

(a) The third person-C produces PAT<AID_{Null} | AID_C> using the MakePAT as follows.

AID_{Null} + AID_C + Enabler of AID_C + Enabler of AID_{Null}

→ PAT<AID_{Null} | AID_C>

(b) The third person-C produces PAT<AID_C | AID_B> using the TransPAT as follows.

PAT<AID_{Null} | AID_B> + PAT<AID_{Null} | AID_C>

+ Enabler of AID_{Null} + Enabler of AID_C

→ PAT<AID_C | AID_B>

[0249] As a result of the above described operation (2)(b), the third person-C obtains PAT<AID_C | AID_B> so that accesses to the member-B become possible.

[0250] For this reason, in this fifth embodiment, it is made impossible for the holder of PAT<AID_{holder} | AID_{member}>

to produce PAT<AID_{Null} | AID_{member}> from this PAT<AID_{holder} | AID_{member}> as long as the holder does not know Enabler of AID_{member}.

[0251] In the third embodiment described above, in order for the PAT holder to produce PAT<AID_{Null} | AID_{member}> without using Enabler of AID_{member}, it is necessary to produce PAT<AID_{holder} | AID_{Null}>.

[0252] To this end, in this fifth embodiment, for the Null-AID described in the third embodiment, the following rule is added:

the Null-AID can be used only as the holder AID of the PAT (the Null-AID cannot be used as the member AID).

That is, PAT<AID_{Null} | AID_{member1}, AID_{member2},, AID_{memberN}> is allowed, but PAT<AID_{holder} | AID_{Null}, AID_{member1}, AID_{member2},, AID_{memberN}> is not allowed.

Each of the secure PAT processing devices and the secure PAT certification authority is additionally equipped with a function for checking whether the Null-AID is contained as the member AID or not. This member AID checking processing is carried out according to Fig. 32 as follows.

(1) Null-AID and PAT are entered (step S6911).

(2) All the member AIDs are taken out from the PAT entered at the step S6911 (step S6913).

(3) Each of the taken out member AIDs is compared with the Null-AID entered at the step S6911 (step S6915).

If all the member AIDs do not completely match with the Null-AID (step S6917 NO, step S6919 NO), the processing proceeds to the MergePAT, SplitPAT or TransPAT processing (Fig. 21 or Fig. 22) (step S6921).

If there is a member AID that completely matches with the Null-AID (step S6917 YES), the processing is terminated.

[0253] Next, with reference to Fig. 33 to Fig. 39, the sixth embodiment of the email access control scheme according to the present invention will be described in detail.

[0254] This sixth embodiment differs from the first embodiment described above in that a link information is added to the AID of Fig. 2 used in the first embodiment, as shown in a part (b) of Fig. 34, while a link information of the AID is set instead of the AID itself that is contained in the 1-to-1 PAT of Fig. 2, as shown in a part (c) of Fig. 34, such that the AID is uniquely identified by the link information.

[0255] Note that such an AID to which the link information is added will be referred to as a link information attached AID, and a 1-to-1 PAT having the link information of the AID will be referred to as a link specifying 1-to-1 PAT. Also, the link information is an information

capable of uniquely identifying the AID, which is given by a kind of data generally known as identifier such as a serial number uniquely assigned to the AID by the CA for example.

[0256] Fig. 33 shows an overall configuration of a communication system in this sixth embodiment.

[0257] In Fig. 33, the CA (Certification Authority) 1 has a right to authenticate OIDs and a right to issue AIDs, and functions to allocate AIDs to users 3.

[0258] The SCS (Secure Communication Service) 5 transfers emails among the users 3, carries out the receiving refusal and the identity judgement and the extraction of the OID according to the need.

[0259] The ADS (Anonymous Directory Service) 7 is a database for managing the AID, the transfer control flag value, the validity period value, and the disclosed information of each user 3. The ADS 7 has a function to generate the PAT from the AID of a searcher and the AID of a registrant who satisfies the search conditions, and issue it to the searcher.

[0260] A series of processing from generating the AID from the OID according to a request from a user until allocating the AID to that user is basically the same as in the first embodiment, except that the link information is to be added, which will now be described with reference to Fig. 34.

[0261] Fig. 34 shows exemplary formats of the OID, the link information attached AID, and the link specifying 1-to-1 PAT. As shown in a part (a) of Fig. 34, the OID is an information comprising an arbitrary character string according to a rule by which the CA 1 can uniquely identify the user and a public key, which is signed by the CA 1.

[0262] Also, as shown in a part (b) of Fig. 34, the link information attached AID is an information comprising fragments of the OID and their position information, redundant character strings, an SCS information given by an arbitrary character string (host name, real domain name, etc.) by which a host or a domain that is operating the SCS 5 can be uniquely identified on the network, and the link information, which is signed by the CA 1.

[0263] Also, as shown in a part (c) of Fig. 34, the link specifying 1-to-1 PAT is an information comprising the transfer control flag, the link information of AID_g, the link information of AID₁, and the validity period, which is signed by the ADS 7 using a secret key of the ADS 7.

[0264] A procedure by which the user 3 requests the link information attached AID to the CA 1 is the same as that of the first embodiment. A procedure by which the CA 1 issues the link information attached AID to the user 3 in response to a request for the AID is also the same as that of the first embodiment.

[0265] Next, the link information attached AID generation processing at the CA will be described with reference to Fig. 35.

[0266] In the procedure of Fig. 35, the CA 1 generates an information of a length equal to the total length L of the OID, and sets this information as a tentative AID

(step S7211). Then, in order to carry out the partial copying of the OID, values of parameters p_i and l_i for specifying a copying region are determined using arbitrary means such as random number generation respectively (step S7213). Here, L is equal to the total length L of the OID, and l_i is an arbitrarily defined value within a range in which a relationship of $0 \leq l_i \leq L$ holds. Then, an information in a range between a position p_i to a position $p_i + l_i$ from the top of the OID is copied to the same positions in the tentative AID (step S7215). In other words, this OID fragment will be copied to a range between a position p_i and a position $p_i + l_i$ from the top of the tentative AID. Then, the values of p_i and l_i are written into a prescribed range in the tentative AID into which the OID has been partially copied, in a form encrypted by an arbitrary means (step S7217). Then, an SCS information given by an arbitrary character string (host name, real domain, etc.) that can uniquely identify a host or a domain that is operating the SCS 5 on the network is written into a prescribed range in the tentative AID into which these values are written (step S7219). Then, the link information is written (step S7220). Then, the tentative AID into which the above character string and the link information are written is signed using a secret key of the CA 1 (step S7221).

[0267] Next, a procedure for registering the AID of a user-B 3 and the disclosed information into the ADS 7 will be described. First, the bidirectional authentication by arbitrary means using the AID of the user-B 3 and the certificate of the ADS 7 is carried out between the user-B 3 who is a registrant and the ADS 7. Then, the user-B 3 transmits the transfer control flag value, the validity period value, and the disclosed information such as interests to the ADS 7. Then, the ADS 7 stores the transfer control flag value, the validity period value, and the entire disclosed information in relation to the AID of the user-B 3 in its storage device. Here, there can be cases where communications between the user-B 3 who is the registrant and the ADS 7 are to be encrypted.

[0268] Next, a procedure by which a user-A 3 searches through the disclosed information that is registered in the ADS 7 will be described. First, the bidirectional authentication by arbitrary means using the AID of the user-A 3 and the certificate of the ADS 7 is carried out between the user-A 3 who is a searcher and the ADS 7. Then, the user-A 3 transmits arbitrary search conditions to the ADS 7. Then, the ADS 7 presents all the received search conditions to its storage device, and extracts the AID of a registrant which satisfies these search conditions. Then, the ADS 7 generates the link specifying 1-to-1 PAT from the link information of the AID of the user-A 3 and the link information of the AID of the registrant who satisfied the search conditions, the transfer control flag value, and the validity period value. Then, the ADS 7 transmits the generated PAT to the user-A 3. Here, there can be cases where communications between the user-A 3 who is a searcher and the ADS 7 are to be encrypted. Note that the link specifying

1-to-1 PAT is generated as a search result of the ADS 7.

[0269] Next, the link specifying 1-to-1 PAT generation processing at the ADS 7 will be described with reference to Fig. 36.

[0270] First, an information of a prescribed length is generated, and this information is set as a tentative PAT (step S7510). Then, the link information of the AID of the user-A 3 who is a searcher and the link information of the AID of the user-B 3 who is a registrant are copied into a prescribed region of the tentative PAT (step S7516). Then, the transfer control flag value and the validity period value are written into respective prescribed regions of the tentative PAT into which the link informations of the AIDs are copied (step S7517). Then, the tentative PAT into which these values are written is signed using a secret key of the ADS 7 (step S7519).

[0271] Next, the transfer control using the link specifying 1-to-1 PAT will be described. The transfer control is a function for limiting accesses to a user who has a proper access right from a third person to whom the PAT has been transferred or who has eavesdropped the PAT (a user who originally does not have the access right).

[0272] The ADS 7 and the user-B 3 of the registrant AID can prohibit a connection to the user-B 3 from a third person who does not have the access right, by setting a certain value in to the transfer control flag of the PAT.

[0273] When the transfer control flag value is set to be 1, the sender's AID is authenticated between the SCS 5 and the sender according to an arbitrary challenge/response process, so that even if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will not be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0274] On the other hand, when the transfer control flag value is set to be 0, no challenge/response process will be carried out between the SCS 5 and the sender, so that if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will also be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0275] Next, the email access control method at the SCS 5 will be described with reference to Fig. 37.

[0276] The sender specifies "[sender's AID]@[real domain of SCS of sender]" in From: line, and "[PAT]@[real domain of SCS of sender]" in To: line.

[0277] The SCS 5 acquires a mail received by an MTA (Message Transfer Agent) such as SMTP (Simple Mail Transfer Protocol), and executes the processing of Fig. 37 as follows.

(1) The signature of the PAT is verified using a public key of the ADS 7 (step S7713).

When the PAT is found to have been altered (step S7715 YES), the mail is discarded and the processing is terminated (step S7716).

When the PAT is found to have been not altered

(step S7715 NO), the following processing (2) is executed.

(2) The search is carried out by presenting the link information of the sender's AID to the PAT (steps S7717, S7720, S7722).

When a link information that completely matches with the link information of the sender's AID is not contained in the PAT (step S7723 NO), the mail is discarded and the processing is terminated (step S7716).

When a link information that completely matches with the link information of the sender's AID is contained in the PAT (step S7723 YES), the following processing (3) is executed.

(3) The validity period value of the PAT is evaluated (steps S7725, S7727).

When the PAT is outside the validity period (step S7727 NO), the mail is discarded and the processing is terminated (step S7716).

When the PAT is within the validity period (step S7727 YES), the following processing (4) is executed.

(4) Whether or not to authenticate the sender is determined by referring to the transfer control flag value of the PAT (steps S7731, S7733).

When the value is 1 (step S7733 YES), the SCS 5 acquires the sender's AID itself and the public key of the sender's AID by presenting the link information to the CA 1, and then the challenge/response authentication between the SCS 5 and the sender is carried out, and the signature of the sender is verified (step S7735). When the signature is valid, the recipient is specified and the PAT is attached (step S7737). When the signature is invalid, the mail is discarded and the processing is terminated (step S7716).

When the value is 0 (step S7733 NO), the recipient is specified and the PAT is attached without executing the challenge/response authentication (step S7737).

[0278] The challenge/response authentication between the SCS 5 and the sender is the same as that for the 1-to-1 PAT described above.

[0279] Next, a method for specifying the recipient at the SCS 5 will be described. First, the SCS 5 carries out the search by presenting the link information of the sender's AID to the PAT, so as to acquire all the link informations which do not completely match the link information of the sender's AID. Then, the search is carried out by presenting all these acquired link informations to the CA 1 so as to acquire the AIDs. All these acquired AIDs will be defined as recipient's AIDs hereafter. Then, for every recipient's AID, the real domain of SCS of recipient is taken out from the recipient's AID. Then, the recipient is specified in a format of "[recipient's AID]@[real domain of SCS of recipient]". Finally, the SCS 5 changes the sender from a format of

"[sender's AID]@[real domain of SCS of sender]" to a format of "sender's AID".

[0280] The method for attaching the PAT at the SCS 5 is the same as that for the 1-to-1 PAT described above.

[0281] Next, a method of receiving refusal with respect to the PAT at the SCS 5 will be described.

[0282] Receiving refusal setting: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user transmits a registration command, his/her own AID, and arbitrary PATs to the SCS 5. Then, the SCS 5 verifies the signature of the received AID. If the signature is invalid, the processing of the SCS 5 is terminated. If the signature is valid, the SCS 5 next verifies the signature of each received PAT using a public key of the ADS. Those PATs with the invalid signature are discarded by the SCS 5. When the signature is valid, the SCS 5 takes out the link information from the received AID, and then carries out the search by presenting the taken out link information to each PAT. For each of those PATs which contain the link information that completely matches with the link information of the received AID, the SCS 5 presents the registration command and the PAT to the storage device such that the PAT is registered into the storage device. Those PATs which do not contain the link information that completely matches with the link information of the received AID are discarded by the SCS 5 without storing them into the storage device. Here, there can be cases where communications between the user and the SCS 5 are to be encrypted.

[0283] Receiving refusal execution: The SCS 5 carries out the search by presenting the PAT to the storage device. When a PAT that completely matches the presented PAT is registered in the storage device, the mail is discarded. When a PAT that completely matches the present PAT is not registered in the storage device, the mail is not discarded.

[0284] Receiving refusal cancellation: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user presents his/her own AID to the SCS 5. Then, the SCS 5 verifies the signature of the received AID. If the signature is invalid, the processing of the SCS 5 is terminated. If the signature is valid, the SCS 5 next takes out the link information from the presented AID, and presents the taken out link information as a search condition to the storage device and acquire all the PATs that contain the presented link information, and then presents all the acquired PATs to the user. Then, the user selects all the PATs for which the receiving refusal is to be cancelled by referring to all the PATs presented from the SCS 5, and transmits all the selected PATs along with a deletion command to the SCS 5. Upon receiving the deletion command and all the PATs for which the receiving refusal is to be cancelled, the SCS 5 presents the deletion command and all the PATs received from the user to the storage device, such that all the received PATs are deleted from the storage

device.

[0285] Note that the method of receiving refusal with respect to the link specifying 1-to-N PAT at the SCS 5 is the same as the method of receiving refusal with respect to the link specifying 1-to-1 PAT described above.

[0286] Next, the judgement of identity will be described with reference to Fig. 38 and Fig. 39.

(1) An initial value of a variable OID_M is defined as a bit sequence with a length equal to the total length L of the OID and all values equal to "0". Also, an initial value of a variable OID_V is defined as a bit sequence with a length equal to the total length of the OID and all values equal to "0" (step S7911).

(2) One link information attached AID is selected from a set of processing target link information attached AIDs, and the following bit processing is carried out (step S7913).

(a) Values of variables AID_M and AID_V are determined according to the position information contained in the link information attached AID (step S7915). Here, AID_M is defined as a bit sequence with a length equal to the total length L of the OID and a value of a position at which the OID information is defined is "1" while a value of a position at which the OID information is not defined is "0" (see Fig. 39). Also, AID_V is defined as a bit sequence with a length equal to the total length L of the OID and a value of a position at which the OID information is defined is an actual value of the OID information while a value of a position at which the OID information is not defined is 0 (see Fig. 39).

(b) AND processing of OID_M and AID_M is carried out and its result is substituted into a variable OVR_M (step S7917).

(c) AND processing of OVR_M and AID_M as well as AND processing of OVR_M and OID_M are carried out and their results are compared (step S7919). When they coincide, OR processing of OID_M and AID_M is carried out and its result is substituted into OID_M (step S7921), while OR processing of OID_V and AID_V is also carried out and its result is substituted into OID_M (step S7923). On the other hand, when they do not coincide, the processing proceeds to the step S7925.

(d) A link information attached AID to be processed next is selected from a set of processing target link information attached AIDs. When at least one another link information attached AID is contained in the set, the steps S7913 to S7923 are executed for that another link information attached AID. When no other link information attached AID is contained in the set, the

processing proceeds to the step S7927.

(e) Values of OID_M and OID_V are outputted (step S7927).

[0287] The value of OID_M that is eventually obtained indicates all positions of the OID information that can be recovered from the set of processing target link information attached AIDs. Also, the value of OID_V that is eventually obtained indicates all the OID information that can be recovered from the set of processing target link information attached AID. In other words, by using the values of OID_M and OID_V , it is possible to obtain the OID albeit probabilistically when the value of OID_V is used as a search condition, and it is possible to quantitatively evaluate a precision of the above search by a ratio OID_M/L with respect to the total length L of the OID.

[0288] As described above, in this sixth embodiment, the CA 1 which is a Trusted Third Party with high secrecy and credibility generates the link information attached AID in which the personal information is concealed, from the OID that contains the highly secret personal information such as name, telephone number, real email address, etc., according to a user request, and issues the AID to the user. By identifying the user by this AID on the communication network as well as in various services provided on the communication network, it becomes possible to provide both the anonymity guarantee and the identity guarantee for the user. In other words, it becomes possible for the user to communicate with another user without revealing the own real name, telephone number, email address, etc., to that another user, and it also becomes possible to disclose the disclosed information to unspecified many through the ADS 7 as will be described below.

[0289] The user registers the disclosed information, that is an information which is supposed to have a low secrecy compared with the personal information at the ADS 7. In the case of searching the disclosed information and the registrant AID, the searcher presents the link information attached AID of the searcher and arbitrary search conditions to the ADS 7. The ADS 7 then extracts the registrant link information attached AID that satisfies these search conditions, and generates the link specifying 1-to-1 PAT from the link information of the AID of the searcher and the link information of the AID of the registrant who satisfied the search conditions, the transfer control flag value, and the validity period value.

[0290] In this link specifying 1-to-1 PAT, the transfer control flag value and the validity period value are set as shown a part (c) of Fig. 34, and by setting up this validity period in advance, it is possible to limit connections from the sender.

[0291] It is also possible to prohibit connections from a third person who does not have the access right, by using the transfer control flag value. Namely, when the transfer control flag value is set to be 1, the sender's AID is authenticated between the SCS 5 and the sender according to an arbitrary challenge/response process,

so that even if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will not be able to make a connection to the registrant of the ADS 7 through the SCS 5. On the other hand, when the transfer control flag value is set to be 0, no challenge/response process will be carried out between the SCS 5 and the sender, so that if the sender gives both the sender's AID and the PAT to another user other than the sender, that another user will also be able to make a connection to the registrant of the ADS 7 through the SCS 5.

[0292] It is also possible to make a connection request to the communication network such that a call for which the recipient is specified by the link specifying 1-to-1 PAT will be received by the recipient's AID or the sender's AID specified by the link information of the link specifying 1-to-1 PAT. In addition, it is also possible to refuse receiving calls with the link specifying 1-to-1 PAT selected by the recipient among calls which are specified by the link specifying 1-to-1 PAT. It is also possible to cancel the receiving refusal of the calls with the link specifying 1-to-1 PAT selected by the recipient. In addition, as a measure against the sender who repeats the personal attack using a plurality of sender's AIDs by taking an advantage of the anonymity, it is possible to judge the identity of the OID from these plurality of sender's AIDs and it is possible to extract that OID at some probability.

[0293] Next, with references to Fig. 40 to Fig. 49, the seventh embodiment of the email access control scheme according to the present invention will be described in detail.

[0294] In contrast to the sixth embodiment described above which is directed to the case where a sender and a recipient are set in 1-to-1 correspondence, this seventh embodiment is directed to the case where a sender and recipients are set in 1-to-N correspondence and a generation of a new link specifying 1-to-N PAT and a content change of the existing link specifying 1-to-N PAT can be made by the initiative of a user, similarly as in the second embodiment described above. Here, the sender is either a holder of the PAT or a member of the PAT. Similarly, the recipient is either a holder of the PAT or a member of the PAT.

[0295] As described in the second embodiment, in general, a membership of a group communication (mailing list, etc.) is changing dynamically so that it is necessary for a host of the group communication to manage information on a point of contact such as telephone number, email address, etc., of each member. In contrast, in the case where it is possible to newly generate a 1-to-1 PAT as in the sixth embodiment, the management of a point of contact is difficult. For example, it is difficult to manage the group collectively, and even if it is given to the others for the purpose of the transfer control, it does not function as an address of the group communication such as mailing list.

[0296] In this seventh embodiment, in order to resolve

such a problem, it is made possible to carry out a generation of a new link specifying 1-to-N PAT and a content change or the existing link specifying 1-to-N PAT by the initiative of a user.

[0297] First, the definition of various identifications used in this seventh embodiment will be described with references to Fig. 40 and Fig. 41.

[0298] As shown in a part (a) of Fig. 40, the OID is an information comprising an arbitrary character string (telephone number, email address, etc.) according to a rule by which the CA 1 can uniquely identify the user and a public key, which is signed by the CA 1.

[0299] Also, as shown in a part (b) of Fig. 40, the link information attached AID is an information comprising fragments of the OID and their position information, redundant character strings, an SCS information given by an arbitrary character string (host name, real domain name, etc.) by which a host or a domain that is operating the SCS 5 can be uniquely identified on the network, and a link information, which is signed by the CA 1. Note that the AID may be encrypted at the SCS 5 or the CA 1. The link information is the same as in the sixth embodiment.

[0300] Also, as shown in a part (c) of Fig. 40, the link specifying 1-to-N PAT is an information comprising two or more link informations of AIDs, a holder index, the validity period, the transfer control flag, and a PAT processing device identifier, which is signed using a secret key of the PAT processing device.

[0301] Here, one of the link informations of AIDs is the link information of the holder AID of this PAT, where the change of the information contained in the PAT such as an addition of the link information of AID to the PAT, a deletion of the link information of AID from the PAT, a change of the validity period in the PAT, a change of the transfer control flag value in the PAT, etc., can be made by presenting the link information of the holder AID and a corresponding Enabler to the PAT processing device.

[0302] On the other hand, the link informations of AIDs other than the link information of the holder AID that are contained in the PAT are all link information of member AIDs, where a change of the information contained in the PAT cannot be made even when the link information of the member AID and a corresponding Enabler are presented to the PAT processing device.

[0303] The holder index is a numerical data for identifying the link information of the holder AID, which is defined to take a value 1 when the link information of the holder AID is a top link information of AID in the link specifying AID list formed from the link information of the holder AID and the link informations of the member AIDs, a value 2 when the link information of the holder AID is a second link information of AID from the top of the link specifying AID list, or a value n when the link information of the holder AID is an n-th link information of AID from the top of the link specifying AID list.

[0304] The transfer control flag value is defined to take either 0 or 1 similarly as in the case of the link specifying

1-to-1 PAT.

[0305] The link information of the holder AID is defined to be a link information of AID which is written at a position of the holder index value in the link specifying AID list. The link informations of the member AIDs are defined to be all the link informations of AIDs other than the link information of the holder AID.

[0306] The validity period is defined by any one or combination of the number of times for which the PAT is available, the absolute time (UTC) by which the PAT becomes unavailable, the absolute time (UTC) by which the PAT becomes available, and the relative time (lifetime) since the PAT becomes available until it becomes unavailable.

[0307] The identifier of a PAT processing device (or a PAT processing object on the network) is defined as a serial number of the PAT processing device (or an distinguished name of the PAT processing object on the network). The secret key of the PAT processing device (or the PAT processing object on the network) is defined to be uniquely corresponding to the identifier.

[0308] Also, in this second embodiment, an Enabler is introduced as an identifier corresponding to the AID. As shown in Fig. 41, the Enabler is an information comprising a character string uniquely indicating that it is an Enabler and a link information attached AID itself, which is signed by the CA 1.

[0309] Next, the operations for a generation of a new PAT and a content change of the existing PAT will be described. Here, the following operations are defined at a secure PAT processing device on the communication terminal or a PAT processing object on the CA or on a network which is properly requested from the CA (which will also be referred to as a PAT processing device hereafter). These operations are similar to those of the second embodiment described above so that they will be described by referring to Fig. 10 to Fig. 13 but it is assumed that each occurrence of AID in Fig. 10 to Fig. 13 should be replaced by the link information of AID in the following.

1. Editing of link specifying AID list:

A link specifying AID list, which is a list of link informations of AIDs contained in the PAT, is edited using link information attached AIDs and Enabler. Else, the link specifying AID list is newly generated.

2. Setting of the validity period and the transfer control flag:

The validity period value and the transfer control flag value contained in the PAT are changed using a link information attached AID and Enabler. Also, a new validity period value and a new transfer control flag value are set in the newly generated link specifying AID list.

[0310] A user who presented the holder AID and the Enabler corresponding to this holder AID to the PAT processing device can edit the list of link informations of

AIDs contained in the PAT. In this case, the following processing rules are used.

(1) Generating a new PAT (MakePAT) (see Fig. 10):

The link specifying AID list (LALIST<(link)holder AID | (link)member AID₁, (link)member AID₂,, (link)member AID_n>) where (link)AID_x denotes the link information of AID_x is newly generated, and the validity period value and the transfer control flag value are set with respect to the generated LALIST.

(link)AID_A + (link)AID_B + Enabler of AID_B
+ Enabler of AID_A
→ LALIST<(link)AID_A | (link)AID_B>
LALIST<(link)AID_A | (link)AID_B> + Enabler of AID_A
+ validity period value
+ transfer control flag value
→ PAT<(link)AID_A | (link)AID_B>

(2) Merging PATs (MergePAT) (see Fig. 11):

A plurality of LALISTs of the same holder AID are merged and the validity period value and the transfer control flag value are set with respect to the merged LALIST.

LALIST<(link)AID_A | (link)AID_{B1}, (link)AID_{B2},>
+ LALIST<(link)AID_A | (link)AID_{C1}, (link)AID_{C2},>
+ Enabler of AID_A
→ LALIST<(link)AID_A | (link)AID_{B1}, (link)AID_{B2},, (link)AID_{C1}, (link)AID_{C2},>
LALIST<(link)AID_A | (link)AID_{B1}, (link)AID_{B2},, (link)AID_{C1}, (link)AID_{C2},>
+ Enabler of AID_A + validity period value
+ transfer control flag value
→ PAT<(link)AID_A | (link)AID_{B1}, (link)AID_{B2},, (link)AID_{C1}, (link)AID_{C2},>

(3) Splitting a PAT (SplitPAT) (see Fig. 12):

The LALIST is split into a plurality of LALISTs of the same holder AID, and the respective validity period value and transfer control flag value are set with respect to each one of the split LALISTs.

LALIST<(link)AID_A | (link)AID_{B1}, (link)AID_{B2},, (link)AID_{C1}, (link)AID_{C2},>
+ Enabler of AID_A
→ LALIST<(link)AID_A | (link)AID_{B1}, (link)AID_{B2},>
+ LALIST<(link)AID_A | (link)AID_{C1}, (link)AID_{C2},>
LALIST<(link)AID_A | (link)AID_{C1}, (link)AID_{C2},>
+ Enabler of AID_A + validity period value
+ transfer control flag value
→ PAT<(link)AID_A | (link)AID_{C1}, (link)AID_{C2},>

(4) Changing a holder of a PAT (TransPAT) (see Fig. 13):

The holder AID of the LALIST is changed, and the validity period value and the transfer control flag value are set with respect to the changed LALIST.

LALIST<(link)AID_A | (link)AID_B>
+ LALIST<(link)AID_A | (link)AID_{C1}, (link)AID_{C2},>
+ Enabler of AID_A + Enabler of AID_B
→ LALIST<(link)AID_B | (link)AID_{C1}, (link)AID_{C2},>
LALIST<(link)AID_B | (link)AID_{C1}, (link)AID_{C2},>
+ Enabler of AID_B + validity period value
+ transfer control flag value
→ PAT<(link)AID_B | (link)AID_{C1}, (link)AID_{C2},>

[0311] In the operation for setting the validity period value, in order to permit the setting of the validity period value only to a user who holds both the holder AID and the corresponding Enabler, the following operation is defined.

PAT<(link)AID_A | (link)AID_B> + Enabler of AID_A
 + validity period value

→ PAT<(link)AID_A | (link)AID_B>

[0312] In the operation for setting the transfer control flag value, in order to permit the setting of the transfer control flag value only to a user who holds both the holder AID and the corresponding Enabler, the following operation is defined.

PAT<(link)AID_A | (link)AID_B> + Enabler of AID_A

+ transfer control flag value

→ PAT<(link)AID_A | (link)AID_B>

[0313] Next, with references to Fig. 42 to Fig. 48, the overall system configuration of this seventh embodiment will be described. In Fig. 42 to Fig. 48, the user-A who has AID_A allocated from the CA stores AID_A and Enabler of AID_A in a computer of the user-A, and the input/output devices such as floppy disk drive, CD-ROM drive, communication board, microphone, speaker, etc., are connected. Else, AID_A and Enabler of AID_A are stored in a communication terminal (telephone, cellular phone, etc.) which has a storage device and a data input/output function.

[0314] Similarly, the user-B who has AID_B allocated from the CA stores AID_B and Enabler of AID_B in a computer of the user-B, and the input/output devices such as floppy disk drive, CD-ROM drive, communication board, microphone, speaker, etc., are connected. Else, AID_B and Enabler of AID_B are stored in a communication terminal (telephone, cellular phone, etc.) which has a storage device and a data input/output function.

[0315] In the following, a procedure by which the user-A generates PAT<(link)AID_A | (link)AID_B> will be described.

(1) The user-A acquires AID_B and Enabler of AID_B using any of the following means.

- AID_B and Enabler of AID_B are registered at the ADS 7, and it is waited until the user-A acquires them as a search result (Fig. 42).
- AID_B and Enabler of AID_B are directly transmitted to the user-A by the email, signaling, etc. (Figs. 43, 44).
- AID_B and Enabler of AID_B are stored in a magnetic, optic, or electronic medium such as floppy disk, CD-ROM, MO, IC card, etc., and this medium is given to the user-A. Else, it is waited until the user acquires them by reading this medium (Figs. 45, 46).
- AID_B and Enabler of AID_B are printed on a paper medium such as book, name card, etc.,

and this medium is given to the user-A. Else, it is waited until the user-A acquire them by reading this medium (Figs. 47, 48).

(2) The user-A who has acquired AID_B and Enabler of AID_B by any of the means described in the above (1) issues the MakePAT command to the PAT processing device. This procedure is common to Fig. 42 to Fig. 48, and defined as follows.

(a) The user A requests the issuance of the MakePAT command by setting AID_A, Enabler of AID_A, AID_B, Enabler of AID_B, the validity period value, and the transfer control flag value into the communication terminal of the user-A.
 (b) The communication terminal of the user-A generates the MakePAT command.

(c) The communication terminal of the user-A transmits the generated MakePAT command to the PAT processing device by means such as the email, signaling, etc. (the issuance of the MakePAT command).

(d) The PAT processing device generates PAT<(link)AID_A | (link)AID_B> by processing the received MakePAT command according to Fig. 21 and Fig. 49. More specifically, this is done as follows.

(link)AID_A + (link)AID_B

+ Enabler of AID_B + Enabler of AID_A

→ LALIST<(link)AID_A | (link)AID_B>

LALIST<(link)AID_A | (link)AID_B> + Enabler of AID_A

+ validity period value + transfer control flag value

→ PAT<(link)AID_A | (link)AID_B>

(e) The PAT processing device transmits the generated PAT<(link)AID_A | (link)AID_B> to the communication terminal of the user-A, or to the communication terminal of the user-B according to the need, by means such as the email, signaling, etc.

(f) The communication terminal of the user-A (or the user-B) stores the received PAT<(link)AID_A | (link)AID_B> in the storage device of the communication terminal of the user-A.

[0316] The merging of PATs (MergePAT, Fig. 21, Fig. 49), the splitting of a PAT (SplitPAT, Fig. 22, Fig. 49), and the changing of a holder of a PAT (TransPAT, Fig. 21, Fig. 49) are also carried out by the similar procedure.

[0317] The procedure of MakePAT, MergePAT and TransPAT is similar to that described above with reference to Fig. 21, except that the AID should be replaced by the link information of the AID and the AID list should be replaced by the link specifying AID list. Also, the procedure of SplitPAT is similar to that described above with reference to Fig. 22, except that the AID should be replaced by the link information of the AID and the AID list should be replaced by the link specifying AID list.

[0318] Here, in the procedures of Fig. 21 and Fig. 22, the link specifying AID list generation is carried out according to Fig. 49 as follows. Namely, a buffer length is determined first (step S9011) and a buffer is generated (step S9012). Then, the link information of the holder AID is copied to a vacant region of the generated buffer (step S9017). Then, the link information of the member AID is copied to a vacant region of the resulting buffer (step S9018), and if the next member AID exists (step S9015 YES), the step S9018 is repeated.

[0319] Next, the determination of the link information of the holder AID will be described. Each of the MakePAT, the MergePAT, the SplitPAT, and the TransPAT commands is defined to have two or more arguments, where AID, PAT, or Enabler can be specified as an argument. In this case, the PAT processing device specifies the link information of the holder AID of the PAT to be outputted after executing each command according to the following rules.

Case of the MakePAT:

For the MakePAT command, it is defined that AIDs are to be specified for the first argument to the N-th argument (N = 2, 3,) and Enablers are to be specified for the N+1-th and subsequent arguments. For example, they can be specified as follows.

MakePAT AID₁, AID₂,, AID_N,
 Enabler of AID₁, Enabler of AID₂,
, Enabler of AID_N

The PAT processing device interprets the link information of AID of the first argument of the MakePAT command as the link information the holder AID.

Only when one of the Enablers of the N+1-th and subsequent arguments corresponds to the AID of the first argument, the PAT processing device specifies the link information of this AID (that is the link information of the AID of the first argument) as the link information of the holder AID of the PAT to be outputted after executing the MakePAT command.

Case of the MergePAT:

For the MergePAT command, it is defined that PATs are to be specified for the first argument to the N-th argument (N = 2, 3,) and Enabler is to be specified for the N+1-th argument.

Namely, they can be specified as follows.

MergePAT PAT₁ PAT₂ PAT_N Enabler of AID

The PAT processing device interprets the link information of the holder AID of the PAT of the first argument of the MergePAT command as the link information of the holder AID of the PAT to be outputted after executing the MergePAT command.

Only when the Enabler of the N+1-th argument corresponds to the holder AID of the PAT of the first argument, the PAT processing device specifies the link information of this AID (that is the link information of the holder AID of the PAT of the first argument) as the link information of the holder AID of the PAT to be outputted after executing the MergePAT command.

Case of the SplitPAT:

For the SplitPAT command, it is defined that PAT is to be specified for the first argument, a set of one or more AIDs grouped together by some prescribed symbols (assumed to be parentheses () in this example) are to be specified for the second argument to the N-th argument (N = 3, 4,), and Enabler is to be specified for the N+1-th argument. Namely, they can be specified as follows.

SplitPAT PAT₁ (AID₁₁) (AID₂₁ AID₂₂)
 (AID_{N1} AID_{N2}
 AID_{NM}) Enabler of AID

The PAT processing device interprets the link information of the holder AID of the PAT of the first argument of the SplitPAT command as the link information of the holder AID of the PAT to be outputted after executing the SplitPAT command.

Only when the Enabler of the N+1-th argument corresponds to the holder AID of the PAT of the first argument, the PAT processing device specifies the link information of this AID (that is the link information of the holder AID of the PAT of the first argument) as the link information of the holder AID of the PAT to be outputted after executing the SplitPAT command.

Case of the TransPAT:

For the TransPAT command, it is defined that PATs are to be specified for the first argument and the second argument, an AID is to be specified for the third argument, and Enablers are to be specified for the fourth argument and the fifth argument. Namely, they can be specified as follows.

TransPAT PAT₁ PAT₂ AID Enabler of AID₁ Enabler of AID₂

The PAT processing device interprets the link

information of AID of the third argument as the link information of the holder AID of the PAT to be outputted after executing the TransPAT command provided that the link information of AID of the third argument of the TransPAT command is contained in the PAT of the second argument.

Only when the Enabler of the fourth argument corresponds to both the PAT of the first argument and the PAT of the second argument and the Enabler of the fifth argument corresponds to the AID of the third argument, the PAT processing device specifies the link information of the AID of the third argument as the link information of the holder AID of the PAT to be outputted after executing the TransPAT command.

Next, the determination of the link informations of the member AIDs will be described. The definitions of the MakePAT, the MergePAT, the SplitPAT, and the TransPAT commands are as described above. The PAT processing device specifies the link informations of the member AIDs of the PAT to be outputted after executing each command according to the following rules.

Case of the MakePAT:

Only when the link information of the holder AID of the PAT to be outputted after executing the MakePAT command is formally determined, the PAT processing device interprets all the link informations of the AIDs of the second and subsequent arguments of the MakePAT command as the link informations of the member AIDs of the PAT to be outputted after executing the MakePAT command.

The PAT processing device specifies only the link informations of those AIDs among all the AIDs of the second and subsequent arguments which correspond to the Enablers specified by the N+1-th and subsequent arguments as the link informations of the member AIDs of the PAT to be outputted after executing the MakePAT command.

Case of the MergePAT:

Only when the link information of the holder AID of the PAT to be outputted after executing the MergePAT command is formally determined, the PAT processing device specifies the link informations of the member AIDs of all the PATs specified by the first to N-th arguments of the MergePAT as the link informations of the member AIDs of the PAT to be outputted after executing the MergePAT command.

Case of the SplitPAT:

Only when the link information of the holder AID of the PAT to be outputted after executing the SplitPAT command is formally determined, the PAT processing device specifies the link information of the member AID of the PAT specified by the first argument of the SplitPAT command as the link information of the member AID of the PAT to be outputted after executing the SplitPAT command. At this

point, the link informations of the member AIDs are distributed into different PATs in units of parentheses (). For example, in the case of:

```
SplitPAT PAT (AID11) (AID21 AID22)
..... (AIDN1 AIDN2 .....
AIDNM) Enabler of AID
```

the link informations of (AID₁₁), (AID₂₁ AID₂₂) and (AID_{N1} AID_{N2} AID_{NM}) will be the link informations of the member AIDs of different PATs having a common link information of holder AID.

Case of TransPAT:

Only when the link information of the holder AID of the PAT to be outputted after executing the TransPAT command is formally determined, the PAT processing device specifies all the link informations of the member AIDs remaining after excluding the link information of the member AID that is scheduled to be a new holder AID from all the link informations of the member AIDs of the PAT specified by the first argument of the TransPAT command and the link informations of the member AIDs of the PAT specified by the second argument as the link informations of the member AIDs of the PAT to be outputted after executing the TransPAT command.

The verification of the properness of the Enabler in this seventh embodiment is the same as described above with reference to Fig. 24. Also, this verification of the properness of the Enabler is common to the MakePAT, the MergePAT, the SplitPAT and the TransPAT.

[0320] Next, the eighth embodiment of the email access control scheme according to the present invention will be described in detail.

[0321] In this eighth embodiment, the OID is given by a real email address.

[0322] The PAT is an information comprising two or more real email addresses, the holder index, the validity period, the transfer control flag and the PAT processing device identifier (or the identifier of the PAT processing object on the network), which is signed using a secret key of the PAT processing device (or the PAT processing object on the network).

[0323] Here, one of the real email addresses is a holder email address of this PAT, where the change of the information contained in the PAT such as an addition of email address to the PAT, a deletion of email address from the PAT, a change of the validity period in the PAT, a change of the transfer control flag value in the PAT, etc., can be made by presenting the holder email address and an Enabler containing the holder email address to the PAT processing device (or the PAT processing object on the network).

[0324] On the other hand, the email addresses other than the holder email address that are contained in the PAT are all member email addresses, where a change

of the information contained in the PAT cannot be made even when the member email address and an Enabler containing the member email address are presented to the PAT processing device (or the PAT processing object on the network).

[0325] The holder index is a numerical data for identifying the holder email address, which is defined to take a value 1 when the holder email address is a top email address in the email address list formed from the holder email address and the member email addresses, a value 2 when the holder email address is a second email address from the top of the email address list, or a value n when the holder email address is an n-th email address from the top of the email address list.

[0326] The transfer control flag value is defined to take either 0 or 1.

[0327] The holder email address is defined to be a real email address which is written at a position specified by the holder index in the email address list. The member email addresses are defined to be all the email addresses other than the holder email address.

[0328] The validity period is defined by any one or combination of the number of times for which the PAT is available, the absolute time (UTC) by which the PAT becomes unavailable, the absolute time (UTC) by which the PAT becomes available, and the relative time (lifetime) since the PAT becomes available until it becomes unavailable.

[0329] The identifier of the PAT processing device (or the PAT processing object on the network) is defined as a serial number of the PAT processing device (or an distinguished name of the PAT processing object on the network). The secret key of the PAT processing device (or the PAT processing object on the network) is defined to be uniquely corresponding to the identifier.

[0330] Also, in this eighth embodiment, an Enabler is defined as an identifier corresponding to the real email address. The Enabler is an information comprising a character string uniquely indicating that it is an Enabler and a real email address itself, which is signed using the secret key of the PAT processing device or the PAT processing object on the network.

[0331] The generation of the PAT in this eighth embodiment is carried out as follows.

[0332] Here, a directory will be described as an example of the PAT processing object on the network. The directory manages the real email address and the disclosed information of the user in correspondence, and outputs the PAT upon receiving the search conditions presented from an arbitrary user.

[0333] The user transmits the real email address and the search conditions to the directory. Then, the directory acquires all the real email addresses which uniquely correspond to the disclosed information that satisfies these search conditions. Then, the directory generates a real email address list from the real email address of the user who presented the search conditions and all the real email addresses acquired as a

search result. Then, the directory appends the holder index value, the validity period value, the transfer control flag value, and the distinguished name of the directory to the real email address list. Finally, the directory signs the resulting data using a secret key of the directory, and transmits it as the PAT to the user who presented the search conditions.

[0334] Next, the email access control in this eighth embodiment is carried out as follows.

[0335] The sender specifies the real email address of the sender in From: line, and "[PAT]@[real domain of sender]" in To: line of a mail.

[0336] The SCS acquires an email received by an MTA (Message Transfer Agent) such as SMTP (Simple Mail Transfer Protocol), and carries out the authentication by the following procedure.

(1) The signature of the PAT is verified using the public key of the PAT.

When the PAT is found to have been altered, the email is discarded and the processing is terminated.

When the PAT is found to have been not altered, the following processing (2) is executed.

(2) The search is carried out by presenting the sender's real email address to the PAT.

When a real email address that completely matches with the sender's real email address is not contained in the PAT, the email is discarded and the processing is terminated.

When a real email address that completely matches with the sender's real email address is contained in the PAT, the following processing (3) is executed.

(3) The validity period value of the PAT is evaluated.

When the PAT is outside the validity period, the email is discarded and the processing is terminated.

When the PAT is within the validity period, the following processing (4) is executed.

(4) Whether or not to authenticate the sender is determined by referring to the transfer control flag value of the PAT.

When the value is 1, the challenge/response authentication between the SCS and the sender is carried out, and the signature of the sender is verified. When the signature is valid, the recipient is specified and the PAT is attached. When the signature is invalid, the email is discarded and the processing is terminated.

When the value is 0, the recipient is specified and the PAT is attached without executing the challenge/response authentication.

[0337] An exemplary challenge/response authentication between the SCS and the sender in this eighth embodiment can be carried out as follows.

[0338] First, the SCS generates an arbitrary informa-

tion such as a timestamp, for example, and transmits the generated information to the sender.

[0339] Then, the sender generates the secret key and the public key, signs the received information using the secret key, and transmits it along with the public key.

[0340] The SCS then verifies the signature of the received information using the public key presented from the sender. When the signature is valid, the recipient is specified and the PAT is attached. When the signature is invalid, the email is discarded and the processing is terminated.

[0341] The specifying of the recipient and the attaching of the PAT at the SCS in this eighth embodiment can be carried out as follows.

[0342] First, the SCS carries out the search by presenting the sender's real email address to the PAT, so as to acquire all the real email addresses which do not completely match the sender's real email address. Then, all these acquired real email addresses are specified as recipient's real email addresses.

[0343] Next, the SCS attaches the PAT to an arbitrary position in the email in order to transmit the PAT to all the recipient's email addresses so as to be able to realize the bidirectional communications. Finally, the SCS gives the email to the MTA.

[0344] The receiving refusal with respect to the PAT at the SCS in this eighth embodiment can be carried out as follows.

[0345] Receiving refusal setting: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user transmits a registration command, his/her own real email address, and arbitrary PATs to the SCS 5. Then, the SCS 5 next verifies the signature of each received PAT using a public key of the ADS. Those PATs with the invalid signature are discarded by the SCS 5. When the signature is valid, the SCS 5 carries out the search by presenting the received real email address to each PAT. For each of those PATs which contain the real email address that completely matches with the received real email address, the SCS 5 presents the registration command and the PAT to the storage device such that the PAT is registered into the storage device. Those PATs which do not contain the real email address that completely matches with the received real email address are discarded by the SCS 5 without storing them into the storage device.

[0346] Receiving refusal execution: The SCS 5 carries out the search by presenting the PAT to the storage device. When a PAT that completely matches the presented PAT is registered in the storage device, the mail is discarded. When a PAT that completely matches the present PAT is not registered in the storage device, the mail is not discarded.

[0347] Receiving refusal cancellation: The bidirectional authentication is carried out by an arbitrary means between the user and the SCS 5. Then, the user presents his/her own real email address to the SCS 5.

Then, the SCS 5 next presents the presented real email address as a search condition to the storage device and acquire all the PATs that contain the presented real email address, and then presents all the acquired PATs to the user. Then, the user selects all the PATs for which the receiving refusal is to be cancelled by referring to all the PATs presented from the SCS 5, and transmits all the selected PATs along with a deletion command to the SCS 5. Upon receiving the deletion command and all the PATs for which the receiving refusal is to be cancelled, the SCS 5 presents the deletion command and all the PATs received from the user to the storage device, such that all the received PATs are deleted from the storage device.

[0348] The editing of the PAT in this eighth embodiment can be carried out as follows.

[0349] The MakePAT, the MergePAT, the SplitPAT, and the TransPAT processings for the PAT using real email addresses as its elements can be obtained from the MakePAT, the MergePAT, the SplitPAT, and the TransPAT processings for the PAT using AIDs as its elements described above, by replacing the AID by the real email address and the Enabler of AID by the Enabler of real email address.

[0350] A Null operator is an information comprising a data which is uniquely indicating that it is Null and which has a format of the real email address, which is signed by the secret key of the PAT processing device or the PAT processing object on the network.

[0351] Similarly, the God operator is an information comprising a data which is uniquely indicating that it is God and which has a format of the real email address, which is signed by the secret key of the PAT processing device or the PAT processing object on the network.

[0352] The Enabler of Null operator is an information comprising a data which is uniquely indicating that it is Enabler and the Null operator itself, which is signed by the secret key of the PAT processing device or the PAT processing object on the network.

[0353] The processings involving the Null operator and the God operator can be obtained from the processings for the PAT using AIDs as its elements described above, by replacing the AID by the real email address, the Enabler of AID by the Enabler of real email address, the Null-AID by the Null operator, the God-AID by the God operator, and the Enabler of Null-AID by the Enabler of Null operator.

[0354] As described, according to the present invention, a PAT is used for verifying the access right of a sender and the email access control among users is carried out when the verification result is valid, so that it becomes possible to disclose the information indicative of characteristics of a user while concealing the true identification of a user and carrying out communications appropriately according to this disclosed information while preventing conventionally possible attacks from a third person. In addition, even when a recipient receives an attack from a sender who maliciously utilizes the

anonymity, damages of a recipient due to that attack can be minimized.

[0355] Also, according to the present invention, the generation and the content change of the personalized access ticket can be made by the initiative of a user by using an AID assigned to each user and an Enabler defined in correspondence to the AID, so that it becomes possible to appropriately manage information such as that of a point of contact of each member of the group communication (mailing list, etc.) which changes dynamically.

[0356] Also, according to the present invention, a Null-AID and an Enabler of Null-AID can be introduced in order to carry out the generation of a new PAT (Make-PAT) and the merging of PATs (MergePAT) without giving the member AID and the Enabler of the member AID to the holder of the PAT, so that it becomes possible to prevent the pretending using the member AID.

[0357] Also, according to the present invention, the Null-AID can be used only as the holder AID of the PAT (the Null-AID cannot be used as the member AID), that is PAT<AID_{Null} | AID_{member1}, AID_{member2},, AID_{memberN} > is allowed, but PAT<AID_{holder} | AID_{Null}, AID_{member1}, AID_{member2},, AID_{memberN} > is not allowed, so that the holder of PAT<AID_{holder} | AID_{member} > cannot produce PAT<AID_{Null} | AID_{member} > from this PAT<AID_{holder} | AID_{member} > as long as the holder does not know Enabler of AID_{member}.

[0358] Also, according to the present invention, a God-AID can be introduced in order to set up a read only attribute to the PAT, so that it becomes possible to fix the participants in the group communication.

[0359] Also, according to the present invention, the link information for uniquely specifying the AID can be introduced and the PAT can be given in terms of the link information such that the PAT does not contain the AID itself, so that it becomes possible to realize the receiving refusal function without using the AID itself.

[0360] It is to be noted that, besides those already mentioned above, many modifications and variations of the above embodiments may be made without departing from the novel and advantageous features of the present invention. Accordingly, all such modifications and variations are intended to be included within the scope of the appended claims.

Claims

1. A method of email access control, comprising the steps of:

receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting

communications between the sender and the receiver; and

controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

2. The method of claim 1, wherein at the controlling step the secure communication service authenticates the personalized access ticket presented by the sender, and refuses a delivery of the email when the personalized access ticket presented by the sender has been altered.

3. The method of claim 2, wherein the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and at the controlling step the secure communication service authenticates the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

4. The method of claim 1, wherein at the receiving step the secure communication service also receives the sender's identification presented by the sender along with the personalized access ticket, and at the controlling step the secure communication service checks whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuses a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

5. The method of claim 1, wherein the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and at the controlling step the secure communication service checks the validity period contained in the personalized access ticket presented by the sender and refuses a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

6. The method of claim 5, wherein the validity period of the personalized access ticket is set by a trusted third party.

7. The method of claim 1, further comprising the step of:

issuing the personalized access ticket to the sender at a directory service for managing an

- identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions. 5 10
8. The method of claim 1, further comprising the step of:
- registering in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service; wherein the controlling step the secure communication service refuses a delivery of the email from the sender when the personalized access ticket presented by the sender is registered therein in advance at the registering step. 15 20 25
9. The method of claim 8, further comprising the step of:
- deleting the personalized access ticket registered at the secure communication service upon request from the specific registrant who registered the personalized access ticket at the registering step. 30 35
10. The method of claim 1, wherein the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service, and at the controlling step, when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the secure communication service authenticates the sender's identification presented by the sender and refuses a delivery of the email when an authentication of the sender's identification fails. 40 45
11. The method of claim 10, wherein the authentication of the sender's identification is realized by a challenge/response procedure between the sender and the secure communication service. 50
12. The method of claim 10, wherein the transfer control flag of the personalized access ticket is set by a trusted third party. 55
13. The method of claim 1, wherein the sender's identification and the recipient's identification in the personalized access ticket are given by real email addresses of the sender and the recipient.
14. The method of claim 1, wherein the sender's identification and the recipient's identification in the personalized access ticket are given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority.
15. The method of claim 14, wherein the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority using a secret key of the certification authority.
16. The method of claim 14, wherein the official identification of each user is a character string uniquely assigned to each user by the certification authority and a public key of each user which are signed by a secret key of the certification authority.
17. The method of claim 14, further comprising the step of:
- probabilistically identifying an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender. 30 35
18. The method of claim 1, wherein an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified are defined, and the sender's identification and the recipient's identification in the personalized access ticket are given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient.
19. The method of claim 1, wherein the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority.
20. The method of claim 18, further comprising the step of:

- probabilistically identifying an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.
21. The method of claim 1, wherein the personalized access ticket contains a single sender's identification and a single recipient's identification in 1-to-1 correspondence.
22. The method of claim 1, wherein the personalized access ticket contains a single sender's identification and a plurality of recipient's identifications in 1-to-N correspondence, where N is an integer greater than 1.
23. The method of claim 22, wherein one identification among the single sender's identification and the plurality of recipient's identifications is a holder identification for identifying a holder of the personalized access ticket while other identifications among the single sender's identification and the plurality of recipient's identifications are member identifications for identifying members of a group to which the holder belongs.
24. The method of claim 23, further comprising the step of:
 issuing an identification of each user and an enabler of the identification of each user indicating a right to change the personalized access ticket containing the identification of each user as the holder identification, to each user at a certification authority, such that prescribed processing on the personalized access ticket can be carried out at a secure processing device only by a user who presented both the holder identification contained in the personalized access ticket and the enabler corresponding to the holder identification to the secure processing device.
25. The method of claim 24, wherein the certification authority issues the enabler of the identification of each user as an information indicating that it is the enabler and the identification of each user itself which are signed by a secret key of the certification authority.
26. The method of claim 24, wherein the prescribed processing includes a generation of a new personalized access ticket, a merging of a plurality of personalized access tickets, a splitting of one personalized access ticket into a plurality of personalized access tickets, a changing of the holder of the personalized access ticket, changing of a validity period of the personalized access ticket, and a changing of a transfer control flag of the personalized access ticket.
27. The method of claim 26, wherein a special identification and a special enabler corresponding to the special identification which are known to all users are defined such that the generation of a new personalized access ticket and the changing of the holder of the personalized access ticket can be carried out by the holder of the personalized access ticket by using the special identification and the special enabler without using an enabler of a member identification.
28. The method of claim 27, wherein the special identification is defined to be capable of being used only as the holder identification of the personalized access ticket.
29. The method of claim 26, wherein a special identification which is known to all users is defined such that a read only attribute can be set to the personalized access ticket by using the special identification.
30. The method of claim 1, wherein at the controlling step, when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the secure communication service takes out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, converts the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and gives the mail after conversion to the mail transfer function by attaching the personalized access ticket.
31. A method of email access control, comprising the steps of:
 defining an official identification of each user by which each user is uniquely identifiable by a certification authority, and an anonymous identification of each user containing at least one fragment of the official identification; and
 identifying each user by the anonymous identification of each user in communications for emails on a communication network.
32. The method of claim 31, wherein the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the

certification authority using a secret key of the certification authority.

33. The method of claim 31, wherein the official identification of each user is a character string uniquely assigned to each user by the certification authority and a public key of each user which are signed by a secret key of the certification authority.

34. The method of claim 31, further comprising the steps of:

receiving a personalized access ticket containing a sender's anonymous identification and a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

35. The method of claim 34, further comprising the step of:

probabilistically identifying an identity of the sender at the secure communication service by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

36. The method of claim 31, wherein the defining step also defines a link information of each anonymous identification by which each anonymous identification can be uniquely identified, and each anonymous identification also contains the link information of each anonymous identification.

37. The method of claim 36, wherein the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority.

38. The method of claim 36, further comprising the steps of:

receiving a personalized access ticket containing a link information of a sender's anonymous identification and a link information of a recipient's anonymous identification in correspondence, which is presented by a sender who

wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, at a secure communication service for connecting communications between the sender and the receiver; and

controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket at the secure communication service.

39. The method of claim 38, further comprising the step of:

probabilistically identifying an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

40. A communication system realizing email access control, comprising:

a communication network to which a plurality of user terminals are connected; and a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

41. The system of claim 40, wherein the secure communication service device authenticates the personalized access ticket presented by the sender, and refuses a delivery of the email when the personalized access ticket presented by the sender has been altered.

42. The system of claim 41, further comprising:

a secure processing device for issuing the personalized access ticket which is signed by a secret key of the secure processing device; wherein the secure communication service device authenticates the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure process-

ing device.

43. The system of claim 40, wherein the secure communication service device also receives the sender's identification presented by the sender along with the personalized access ticket, checks whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuses a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.
44. The system of claim 40, wherein the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the secure communication service device checks the validity period contained in the personalized access ticket presented by the sender and refuses a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.
45. The system of claim 44, further comprising:
a trusted third party for setting the validity period of the personalized access ticket.
46. The system of claim 40, further comprising:
a directory service device for managing an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and issuing the personalized access ticket to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.
47. The system of claim 40, wherein the secure communication service device registers in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, and refuses a delivery of the email from the sender when the personalized access ticket presented by the sender is registered therein in advance.
48. The system of claim 47, wherein the secure communication service device deletes the personalized

access ticket registered therein upon request from the specific registrant who registered the personalized access ticket.

49. The system of claim 40, wherein the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the secure communication service device authenticates the sender's identification presented by the sender and refuses a delivery of the email when an authentication of the sender's identification fails.
50. The system of claim 49, wherein the authentication of the sender's identification is realized by a challenge/response procedure between the sender and the secure communication service device.
51. The system of claim 49, further comprising a trusted third party for setting the transfer control flag of the personalized access ticket.
52. The system of claim 40, wherein the sender's identification and the recipient's identification in the personalized access ticket are given by real email addresses of the sender and the recipient.
53. The system of claim 40, further comprising:
a certification authority device for issuing an anonymous identification of each user which contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by the certification authority device;
wherein the sender's identification and the recipient's identification in the personalized access ticket are given by anonymous identifications of the sender and the recipient.
54. The system of claim 53, wherein the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority device using a secret key of the certification authority device.
55. The system of claim 53, wherein the official identification of each user is a character string uniquely assigned to each user by the certification authority device and a public key of each user which are signed by a secret key of the certification authority device.
56. The system of claim 53, wherein the secure com-

munication service device probabilistically identifies an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

57. The system of claim 40, further comprising:

a certification authority device for issuing an anonymous identification of each user which contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by the certification authority device and a link information of each anonymous identification by which each anonymous identification can be uniquely identified; wherein the sender's identification and the recipient's identification in the personalized access ticket are given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient.

58. The system of claim 57, wherein the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority device.

59. The system of claim 57, wherein the secure communication service device probabilistically identifies an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

60. The system of claim 40, wherein the personalized access ticket contains a single sender's identification and a single recipient's identification in 1-to-1 correspondence.

61. The system of claim 40, wherein the personalized access ticket contains a single sender's identification and a plurality of recipient's identifications in 1-to-N correspondence, where N is an integer greater than 1.

62. The system of claim 61, wherein one identification among the single sender's identification and the plurality of recipient's identifications is a holder identification for identifying a holder of the personalized access ticket while other identifications among the single sender's identification and the plurality of recipient's identifications are member identifications for identifying members of a group to which the holder belongs.

63. The system of claim 62, further comprising:

a certification authority device for issuing to each user an identification of each user and an enabler of the identification of each user indicating a right to change the personalized access ticket containing the identification of each user as the holder identification; and a secure processing device at which prescribed processing on the personalized access ticket can be carried out only by a user who presented both the holder identification contained in the personalized access ticket and the enabler corresponding to the holder identification to the secure processing device.

64. The system of claim 63, wherein the certification authority device issues the enabler of the identification of each user as an information indicating that it is the enabler and the identification of each user itself which are signed by a secret key of the certification authority device.

65. The system of claim 63, wherein the prescribed processing includes a generation of a new personalized access ticket, a merging of a plurality of personalized access tickets, a splitting of one personalized access ticket into a plurality of personalized access tickets, a changing of the holder of the personalized access ticket, changing of a validity period of the personalized access ticket, and a changing of a transfer control flag of the personalized access ticket.

66. The system of claim 65, wherein a special identification and a special enabler corresponding to the special identification which are known to all users are defined such that the generation of a new personalized access ticket and the changing of the holder of the personalized access ticket can be carried out by the holder of the personalized access ticket by using the special identification and the special enabler without using an enabler of a member identification.

67. The system of claim 66, wherein the special identification is defined to be capable of being used only as the holder identification of the personalized access ticket.

68. The system of claim 65, wherein a special identification which is known to all users is defined such that a read only attribute can be set to the personalized access ticket by using the special identification.

69. The system of claim 40, wherein when the access right of the sender with respect to the recipient is

verified according to the personalized access ticket, the secure communication service device takes out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, converts the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and gives the mail after conversion to the mail transfer function by attaching the personalized access ticket.

70. A communication system realizing email access control, comprising:

a certification authority device for defining an official identification of each user by which each user is uniquely identifiable by the certification authority device, and an anonymous identification of each user which contains at least one fragment of the official identification; and

a communication network on which each user is identified by the anonymous identification of each user in communications for emails on the communication network.

71. The system of claim 70, wherein the anonymous identification of each user is an information containing the at least one fragment of the official identification of each user which is signed by the certification authority device using a secret key of the certification authority device.

72. The system of claim 70, wherein the official identification of each user is a character string uniquely assigned to each user by the certification authority device and a public key of each user which are signed by a secret key of the certification authority device.

73. The system of claim 70, further comprising:

a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a sender's anonymous identification and a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

74. The system of claim 73, wherein the secure communication service device probabilistically identifies an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

75. The system of claim 70, wherein the certification authority device also defines a link information of each anonymous identification by which each anonymous identification can be uniquely identified, and each anonymous identification also contains the link information of each anonymous identification.

76. The system of claim 75, wherein the link information of each anonymous identification is an identifier uniquely assigned to each anonymous identification by the certification authority device.

77. The system of claim 75, further comprising:

a secure communication service device for connecting communications between the sender and the receiver on the communication network, by receiving a personalized access ticket containing a link information of a sender's anonymous identification and a link information of a recipient's anonymous identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

78. The system of claim 77, wherein the secure communication service device probabilistically identifies an identity of the sender by reconstructing the official identification of the sender while judging identity of a plurality of link informations of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

79. A secure communication service device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to connect communications between the sender and the receiver, by receiving a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a

sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email, and controlling accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket.

80. The secure communication service device of claim 79,

wherein the computer software causes the computer hardware to authenticate the personalized access ticket presented by the sender, and refuse a delivery of the email when the personalized access ticket presented by the sender has been altered.

81. The secure communication service device of claim 80,

wherein the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and the computer software causes the computer hardware to authenticate the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

82. The secure communication service device of claim 79,

wherein the computer software causes the computer hardware to also receive the sender's identification presented by the sender along with the personalized access ticket, check whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the sender, and refuse a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

83. The secure communication service device of claim 79,

wherein the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the computer software causes the computer hardware to check the validity period contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

84. The secure communication service device of claim 79,

wherein the computer software causes the computer hardware to register in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a

specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service device, and refuse a delivery of the email from the sender when the personalized access ticket presented by the sender is registered at the secure communication service device in advance.

85. The secure communication service device of claim 84,

wherein the computer software causes the computer hardware to delete the personalized access ticket registered at the secure communication service device upon request from the specific registrant who registered the personalized access ticket.

86. The secure communication service device of claim 79,

wherein the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service device, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the computer software causes the computer hardware to authenticate the sender's identification presented by the sender and refuse a delivery of the email when an authentication of the sender's identification fails.

87. The secure communication service device of claim 86,

wherein the computer software causes the computer hardware to realize the authentication of the sender's identification by a challenge/response procedure between the sender and the secure communication service device.

88. The secure communication service device of claim 79,

wherein the sender's identification and the recipient's identification in the personalized access ticket are given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority, and the computer software also causes the computer hardware to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.

89. The secure communication service device of claim 79,

wherein an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified are defined, the sender's identification and the recipient's identification in the personalized access ticket are given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient, and the computer software also causes the computer hardware to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.

90. The secure communication service device of claim 79,

wherein when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the computer software causes the computer hardware to take out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, convert the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and give the mail after conversion to the mail transfer function by attaching the personalized access ticket.

91. A secure processing device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to receive a request for a personalized access ticket from a user, and issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is signed by a secret key of the secure processing device.

92. A directory service device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to manage an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a

personal information, in a state which is accessible for search by unspecified many, and issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

93. A certification authority device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to issue to each user an official identification of each user by which each user is uniquely identifiable by the certification authority device, and an anonymous identification of each user which contains at least one fragment of the official identification.

94. A certification authority device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to issue to each user an identification of each user and an enabler of the identification of each user indicating a right to change any personalized access ticket that contains the identification of each user as a holder identification, where the personalized access ticket generally contains a sender's identification and a plurality of recipient's identifications in correspondence, and one of the sender's identification and the recipient's identifications is a holder identification.

95. A secure processing device for use in a communication system realizing email access control, comprising:

a computer hardware; and
a computer software for causing the computer hardware to receive from a user a request for prescribed processing on a personalized access ticket containing a sender's identification and a plurality of recipient's identifications in correspondence, where one of the sender's identification and the recipient's identifications is a holder identification, and execute the prescribed processing on the personalized access ticket when the user presented both the holder

identification contained in the personalized access ticket and an enabler corresponding to the holder identification which indicates a right to change the personalized access ticket containing the identification of the user as the holder identification.

96. A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure communication service device for use in a communication system realizing email access control, the computer readable program code means includes:

first computer readable program code means for causing said computer to receive a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is presented by a sender who wishes to send an email to a recipient so as to specify the recipient as an intended destination of the email; and second computer readable program code means for causing said computer to control accesses between the sender and the recipient by verifying an access right of the sender with respect to the recipient according to the personalized access ticket, so as to connect communications between the sender and the receiver on the communication network.

97. The computer usable medium of claim 96, the second computer readable program code means causes said computer to authenticate the personalized access ticket presented by the sender, and refuse a delivery of the email when the personalized access ticket presented by the sender has been altered.

98. The computer usable medium of claim 97, wherein the personalized access ticket is signed by a secret key of a secure processing device which issued the personalized access ticket, and the second computer readable program code means causes said computer to authenticate the personalized access ticket by verifying a signature of the secure processing device in the personalized access ticket using a public key of the secure processing device.

99. The computer usable medium of claim 96, wherein the first computer readable program code means causes said computer to also receive the sender's identification presented by the sender along with the personalized access ticket, and the second computer readable program code means causes said computer to check whether the sender's identification presented by the sender is contained in the personalized access ticket presented by the

sender and refuse a delivery of the email when the sender's identification presented by the sender is not contained in the personalized access ticket presented by the sender.

100. The computer usable medium of claim 96, wherein the personalized access ticket also contains a validity period indicating a period for which the personalized access ticket is valid, and the second computer readable program code means causes said computer to check the validity period contained in the personalized access ticket presented by the sender and refuse a delivery of the email when the personalized access ticket presented by the sender contains the validity period that has already been expired.

101. The computer usable medium of claim 96, wherein the second computer readable program code means causes said computer to register in advance the personalized access ticket containing an identification of a specific user from which a delivery of emails to a specific registrant is to be refused as the sender's identification and an identification of the specific registrant as the recipient's identification, at the secure communication service device, and refuse a delivery of the email from the sender when the personalized access ticket presented by the sender is registered at the secure communication service device in advance.

102. The computer usable medium of claim 101, wherein the second computer readable program code means causes said computer to delete the personalized access ticket registered at the secure communication service device upon request from the specific registrant who registered the personalized access ticket.

103. The computer usable medium of claim 96, wherein the personalized access ticket also contains a transfer control flag indicating whether or not the sender should be authenticated by the secure communication service device, and when the transfer control flag contained in the personalized access ticket indicates that the sender should be authenticated, the second computer readable program code means causes said computer to authenticate the sender's identification presented by the sender and refuse a delivery of the email when an authentication of the sender's identification fails.

104. The computer usable medium of claim 103, wherein the second computer readable program code means causes said computer to realize the authentication of the sender's identification by a challenge/response procedure between the sender and the secure communication service device.

105. The computer usable medium of claim 96, wherein the sender's identification and the recipient's identification in the personalized access ticket are given by anonymous identifications of the sender and the recipient, where an anonymous identification of each user contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority, and the second computer readable program code means also causes said computer to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender contained in a plurality of personalized access tickets used by the sender.
106. The computer usable medium of claim 96, wherein an anonymous identification of each user that contains at least one fragment of an official identification of each user by which each user is uniquely identifiable by a certification authority and a link information of each anonymous identification by which each anonymous identification can be uniquely identified are defined, the sender's identification and the recipient's identification in the personalized access ticket are given by a link information of the anonymous identification of the sender and a link information of the anonymous identification of the recipient, and the second computer readable program code means also causes said computer to probabilistically identify an identity of the sender by reconstructing the official identification of the sender by judging identity of a plurality of anonymous identifications of the sender corresponding to the link information contained in a plurality of personalized access tickets used by the sender.
107. The computer usable medium of claim 96, wherein when the access right of the sender with respect to the recipient is verified according to the personalized access ticket, the second computer readable program code means causes said computer to take out the recipient's identification from the personalized access ticket by using the sender's identification presented by the sender, convert the mail by using a taken out recipient's identification into a format that can be interpreted by a mail transfer function for actually carrying out a mail delivery processing, and give the mail after conversion to the mail transfer function by attaching the personalized access ticket.
108. A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure processing device for use in a communication system realizing email access control, the computer

readable program code means includes:

first computer readable program code means for causing said computer to receive a request for a personalized access ticket from a user; and
second computer readable program code means for causing said computer to issue the personalized access ticket containing a sender's identification and a recipient's identification in correspondence, which is signed by a secret key of the secure processing device.

109. A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a directory service device for use in a communication system realizing email access control, the computer readable program code means includes:

first computer readable program code means for causing said computer to manage an identification of each registrant and a disclosed information of each registrant which has a lower secrecy than a personal information, in a state which is accessible for search by unspecified many, and
second computer readable program code means for causing said computer to issue a personalized access ticket containing a sender's identification and a recipient's identification in correspondence, to the sender in response to search conditions specified by the sender, by using an identification of a registrant whose disclosed information matches the search conditions as the recipient's identification and the sender's identification specified by the sender along with the search conditions.

110. A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a certification authority device for use in a communication system realizing email access control, the computer readable program code means includes:

first computer readable program code means for causing said computer to issue to each user an official identification of each user by which each user is uniquely identifiable by the certification authority device; and
second computer readable program code means for causing said computer to issue to each user an anonymous identification of each user which contains at least one fragment of the official identification.

111. A computer usable medium having computer read-

able program code means embodied therein for causing a computer to function as a certification authority device for use in a communication system realizing email access control, the computer readable program code means includes:

5

first computer readable program code means for causing said computer to issue to each user an identification of each user; and

second computer readable program code means for causing said computer to issue to each user an enabler of the identification of each user indicating a right to change any personalized access ticket that contains the identification of each user as a holder identification, where the personalized access ticket generally contains a sender's identification and a plurality of recipient's identifications in correspondence, and one of the sender's identification and the recipient's identifications is a holder identification.

10

15

20

112.A computer usable medium having computer readable program code means embodied therein for causing a computer to function as a secure processing device for use in a communication system realizing email access control, the computer readable program code means includes:

25

first computer readable program code means for causing said computer to receive from a user a request for prescribed processing on a personalized access ticket containing a sender's identification and a plurality of recipient's identifications in correspondence, where one of the sender's identification and the recipient's identifications is a holder identification; and

30

35

second computer readable program code means for causing said computer to execute the prescribed processing on the personalized access ticket when the user presented both the holder identification contained in the personalized access ticket and an enabler corresponding to the holder identification which indicates a right to change the personalized access ticket containing the identification of the user as the holder identification.

40

45

50

55

51

FIG. 1

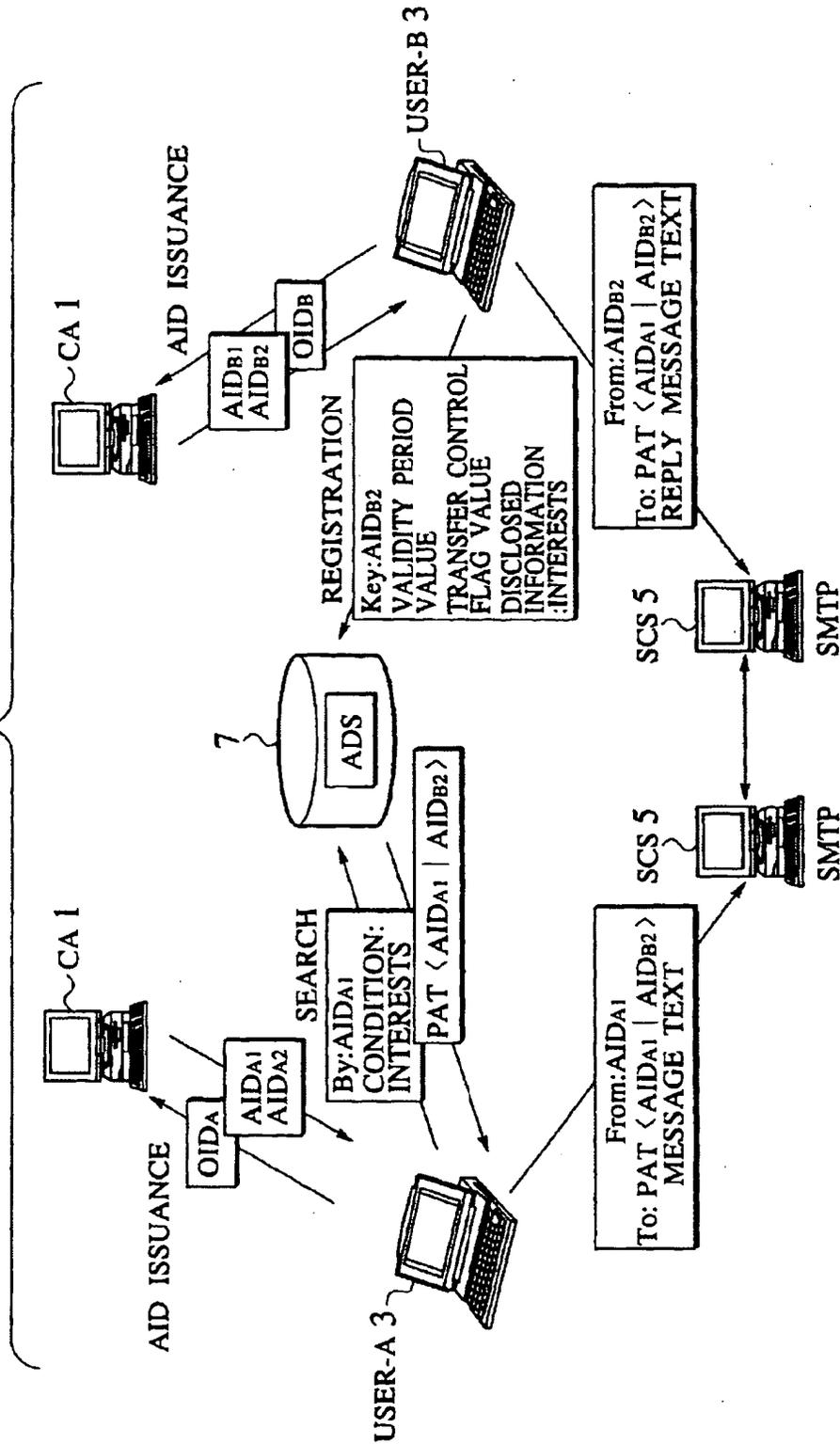
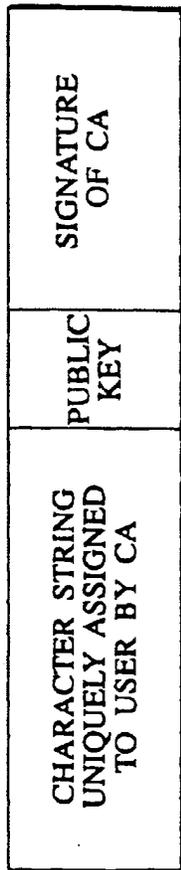
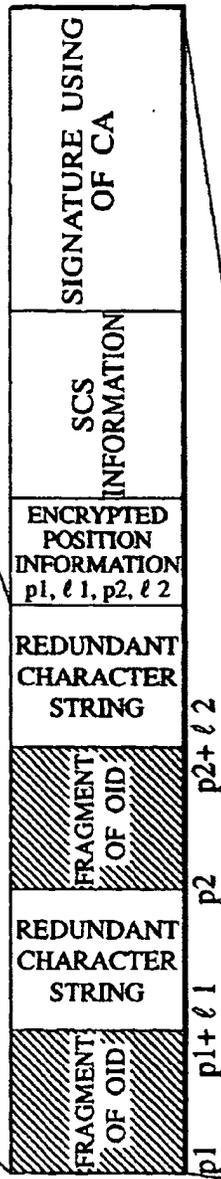


FIG.2

(a) Official Identification:OID



(b) Anonymous Identification:AID



(c) 1-To-1 Personalized Access Ticket:PAT

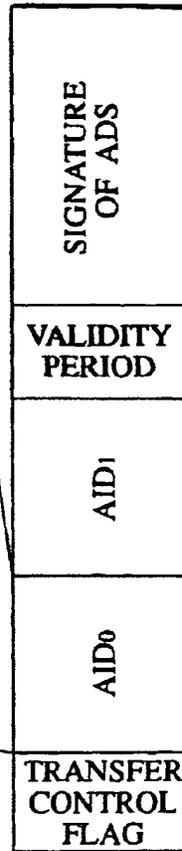


FIG.3

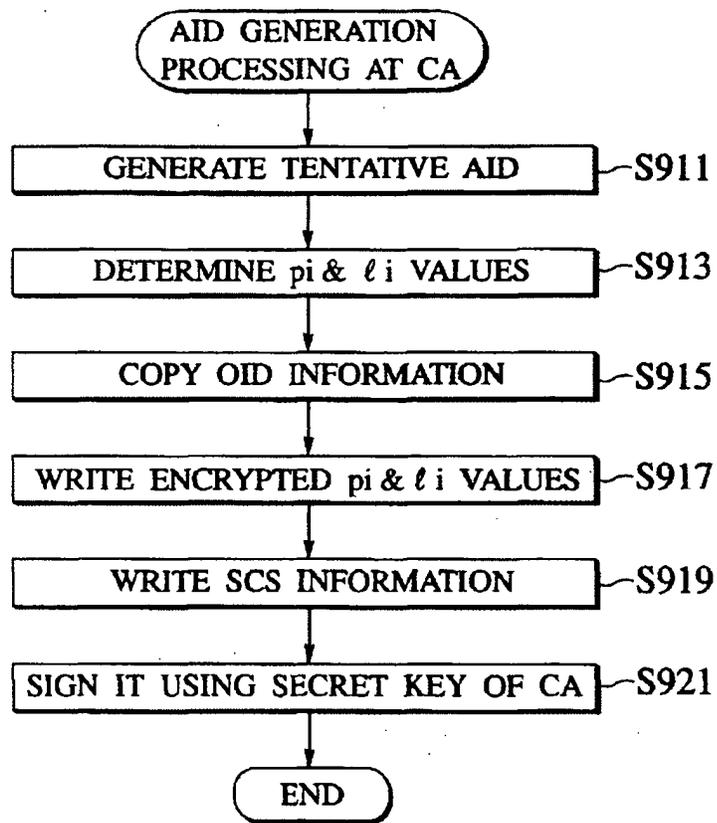


FIG.4

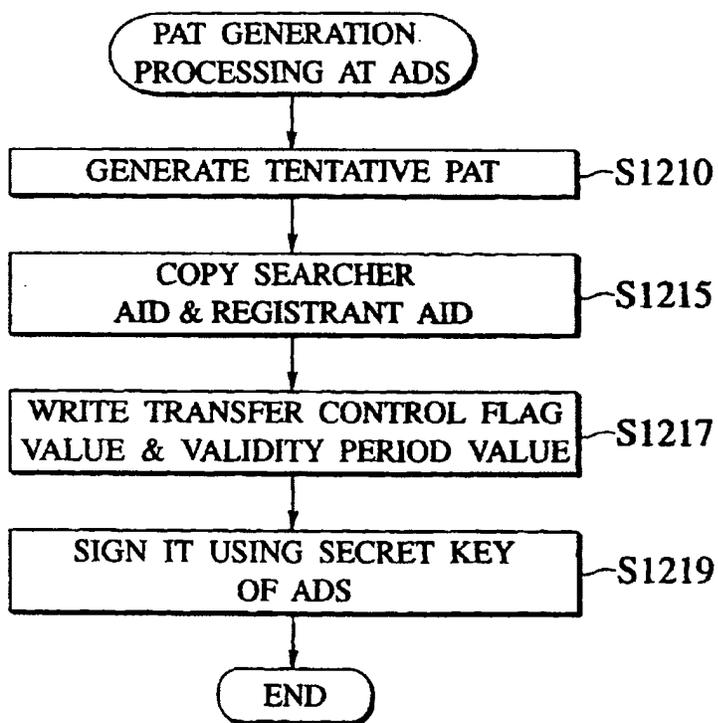


FIG.5

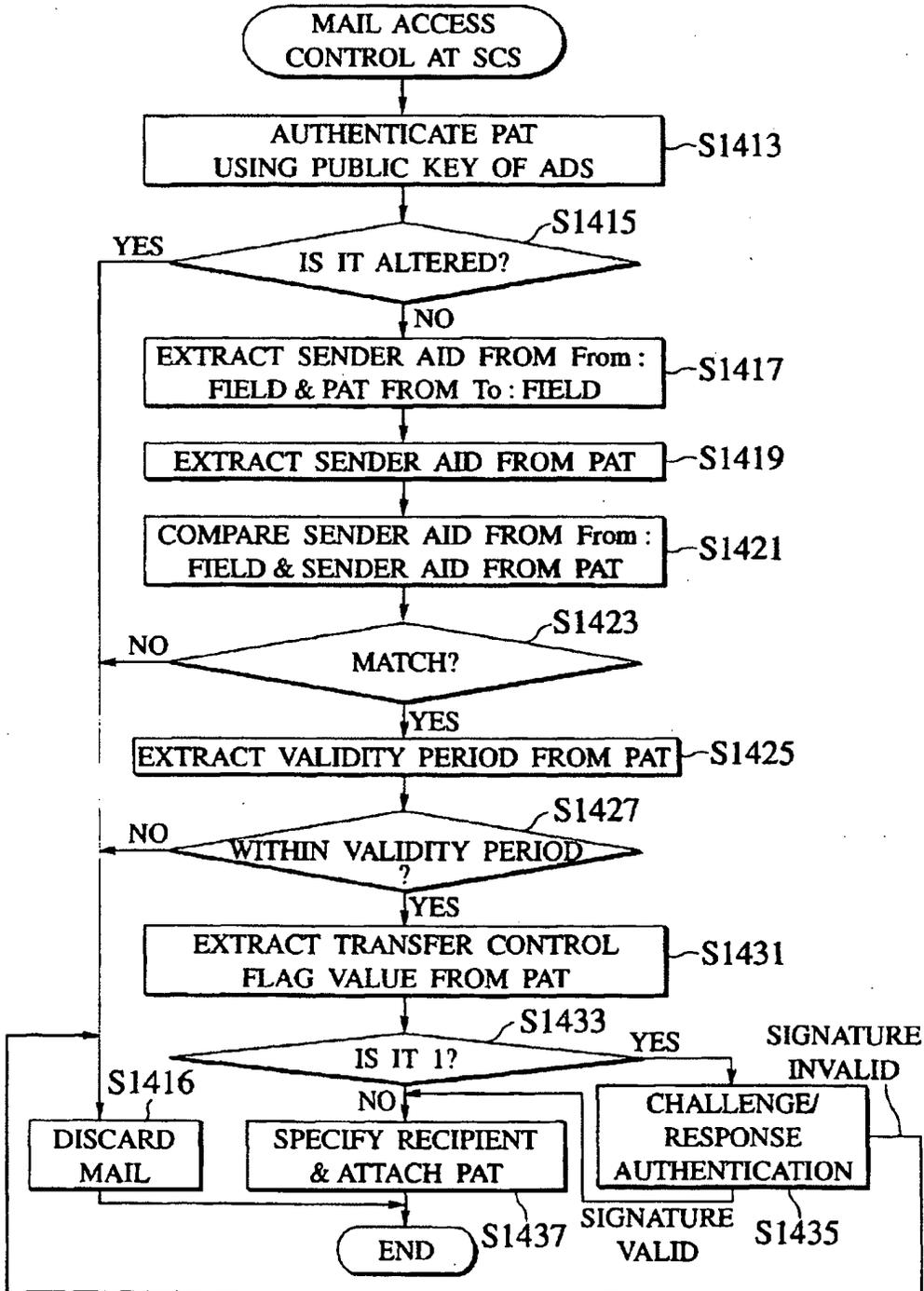


FIG.6

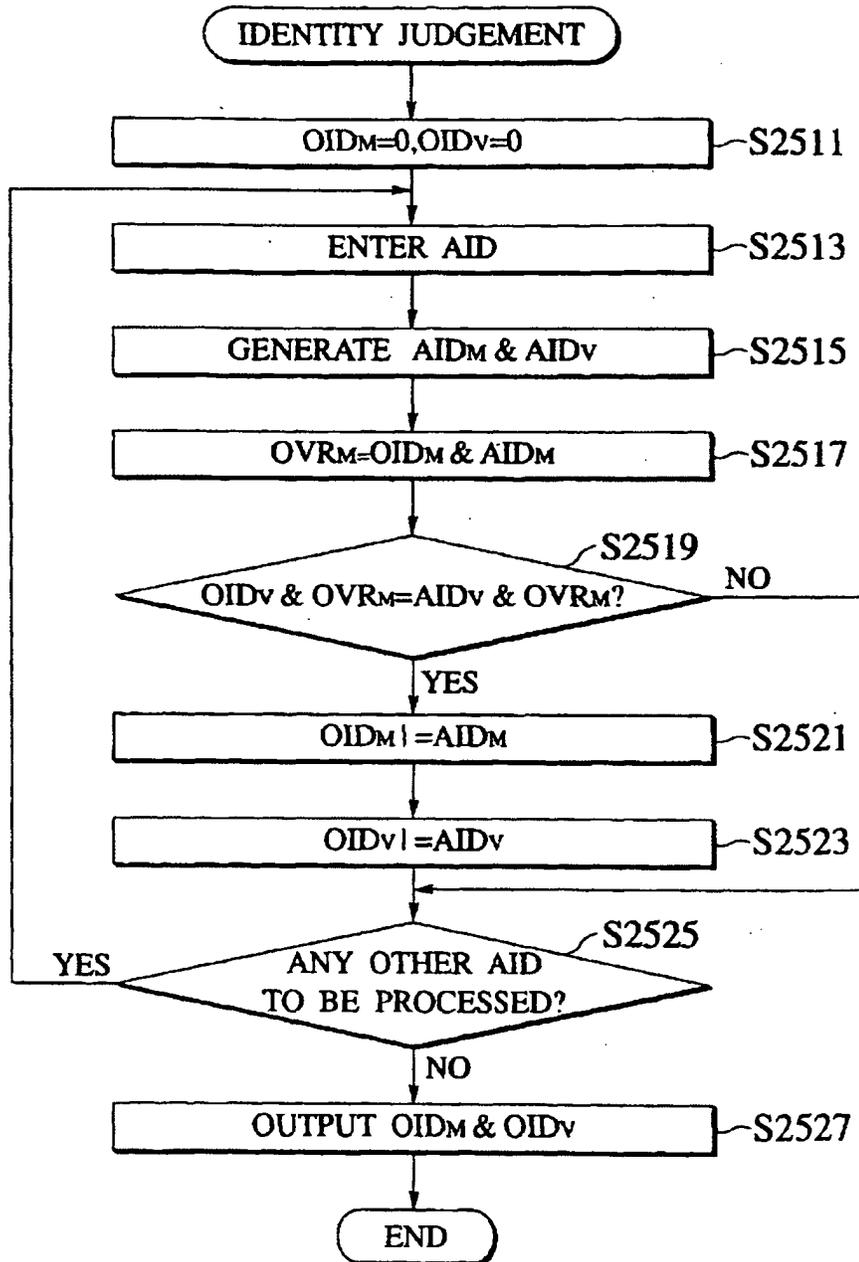


FIG. 7

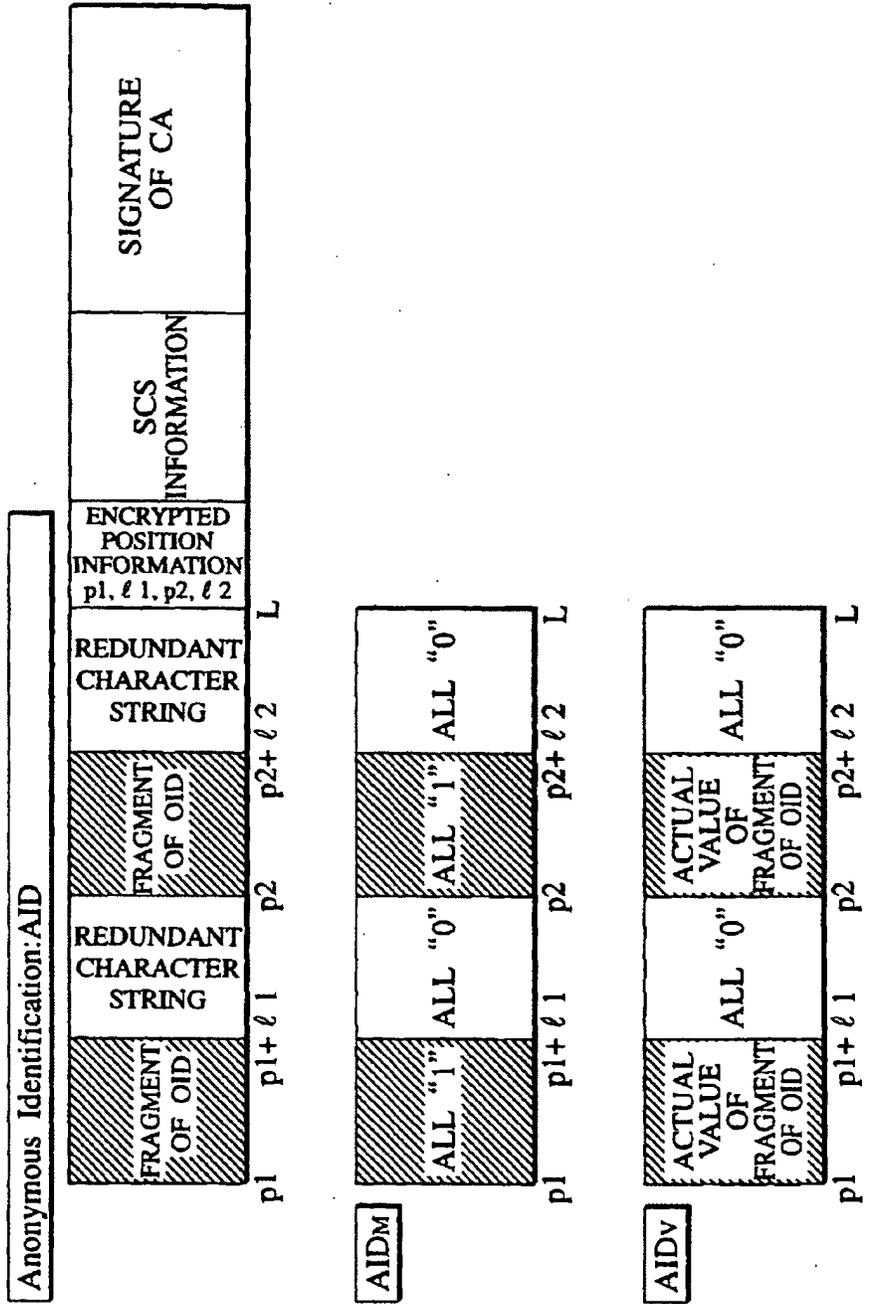


FIG.8

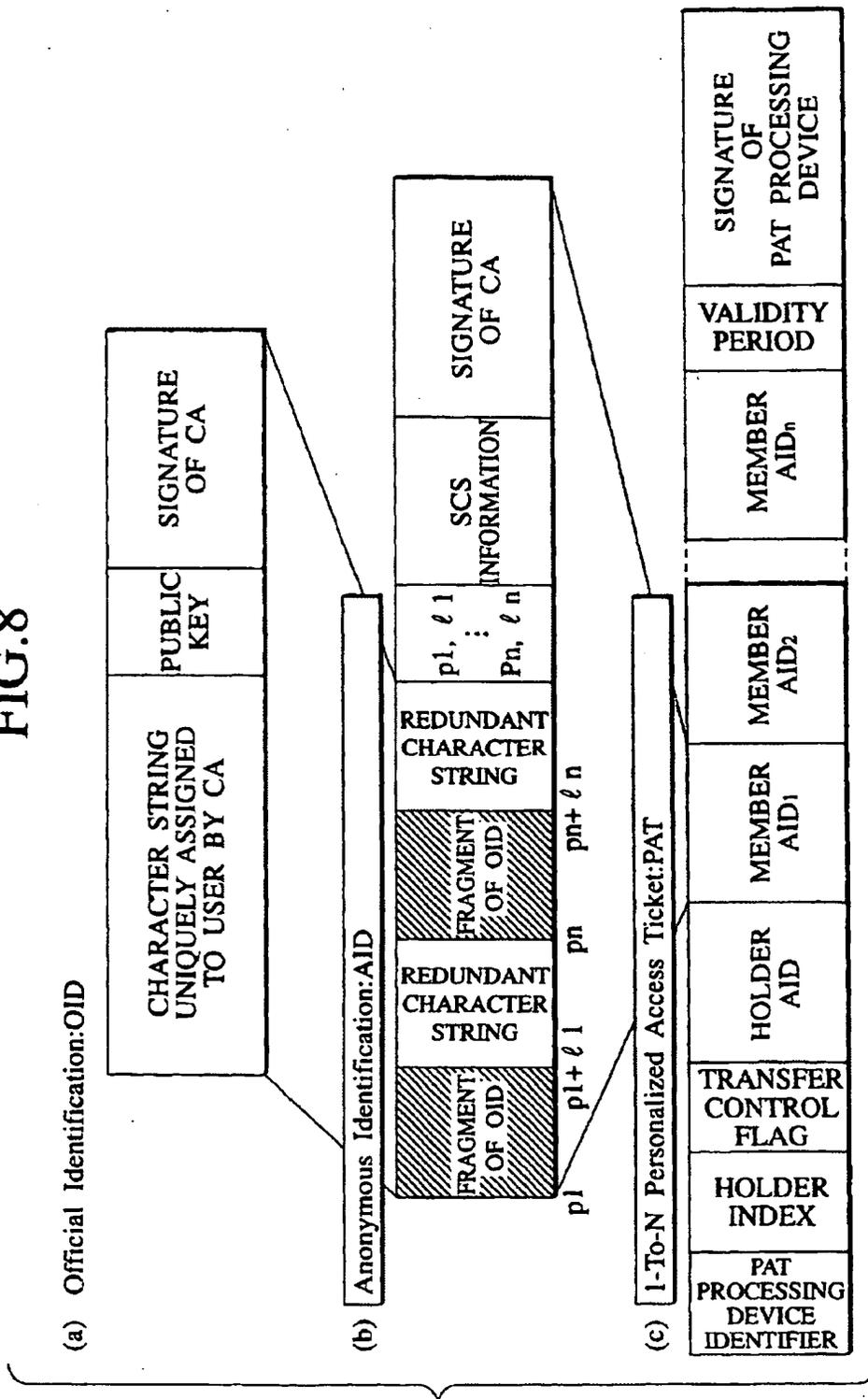


FIG.9

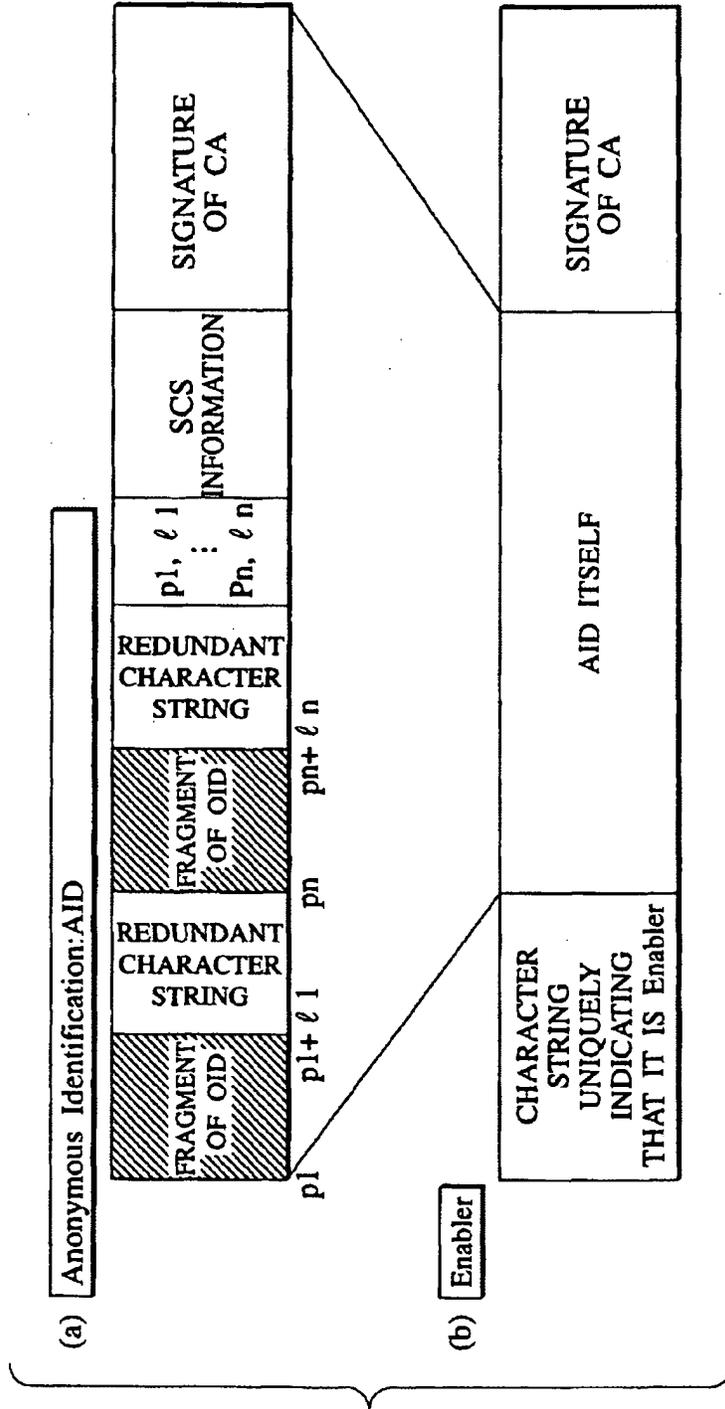


FIG.10

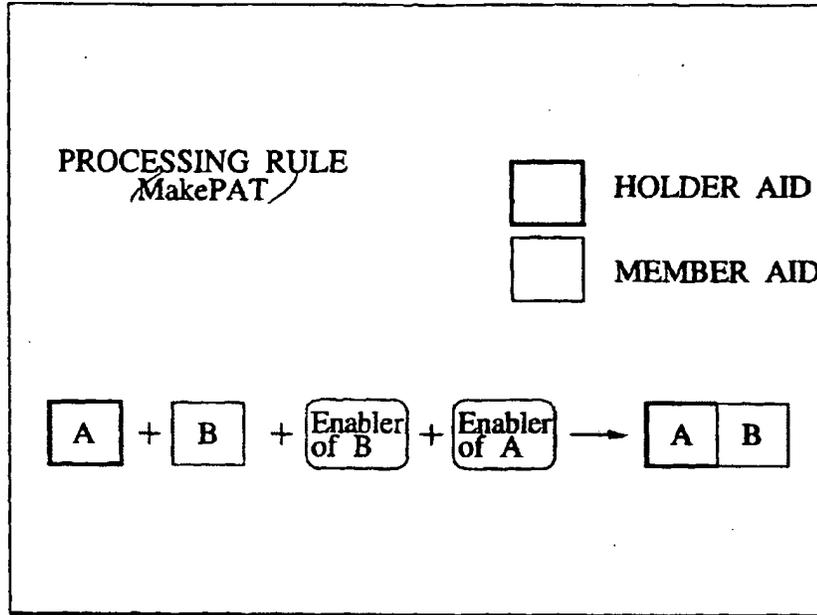


FIG.11

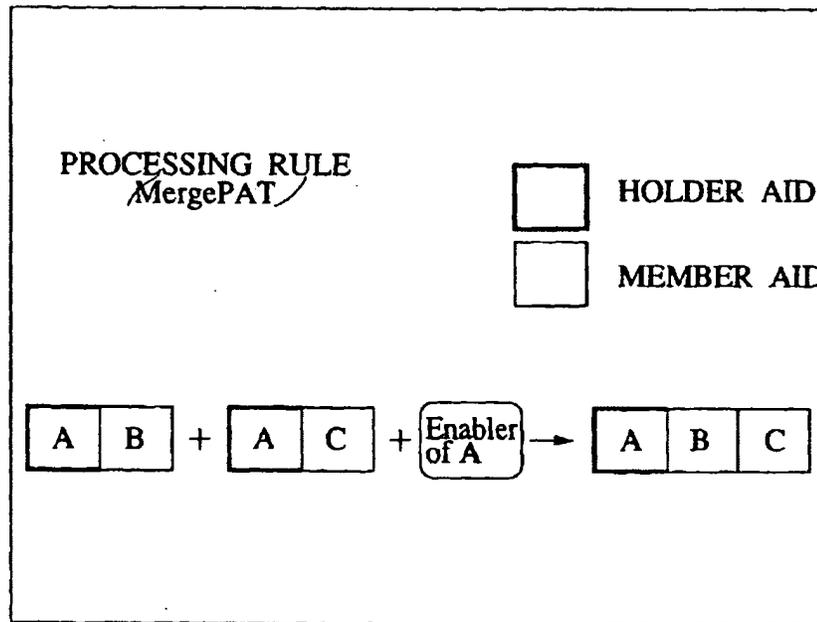


FIG.12

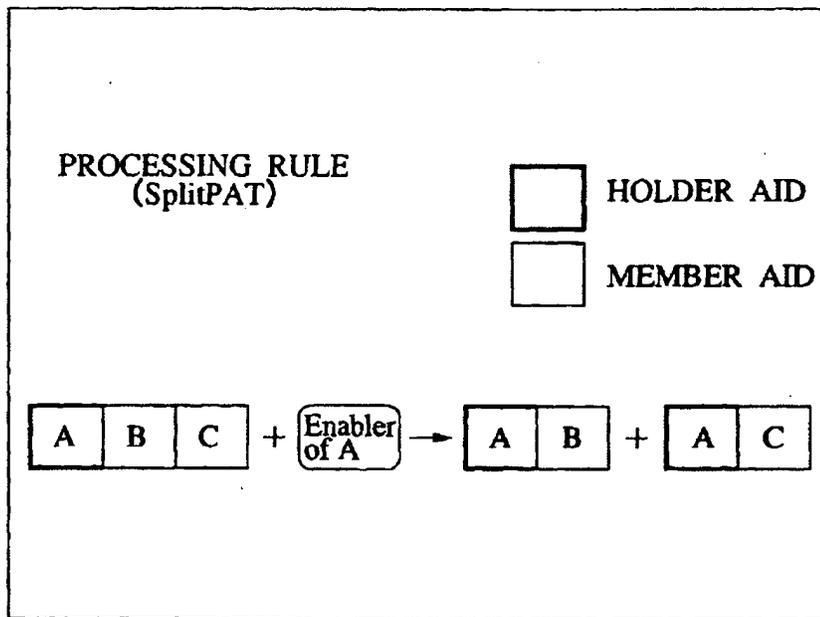


FIG.13

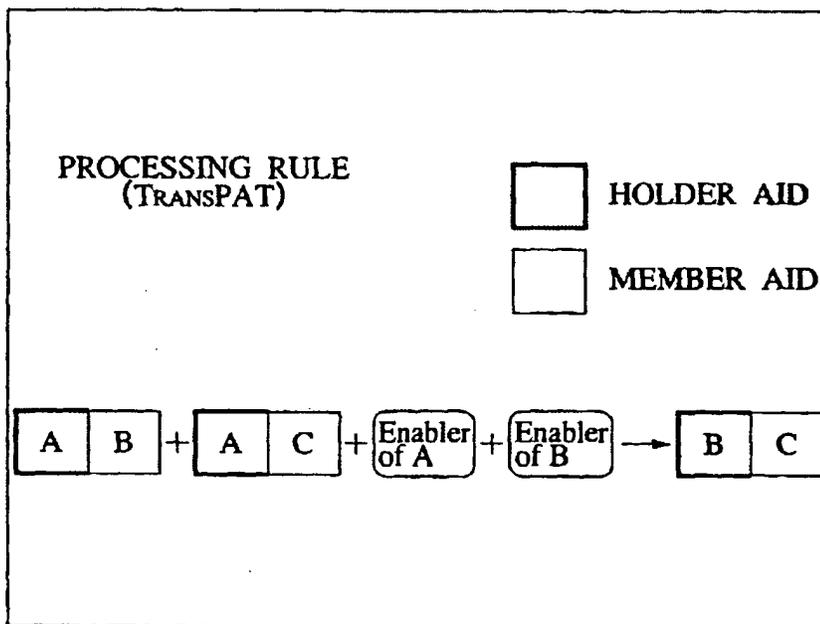


FIG.14

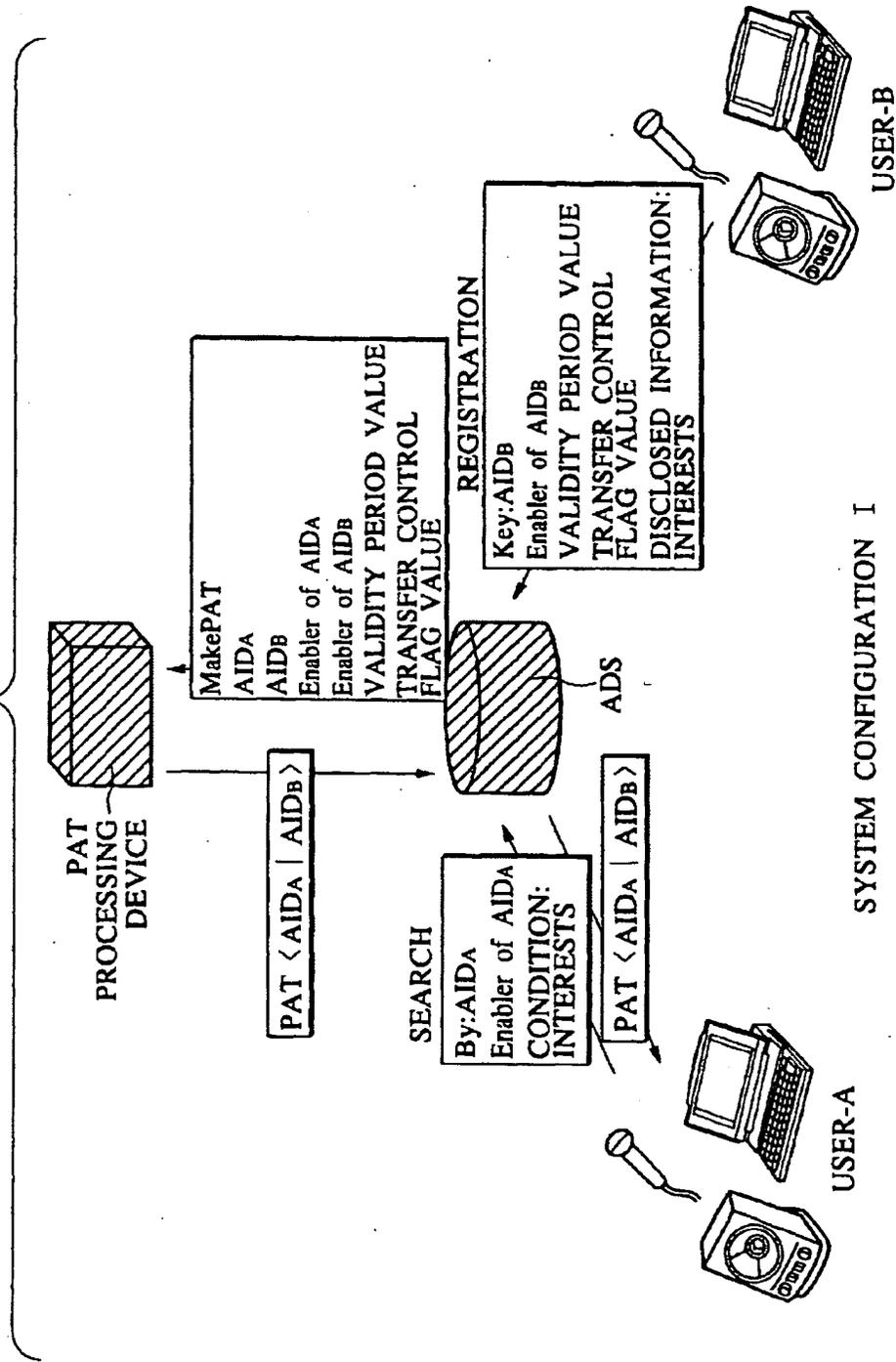


FIG.15

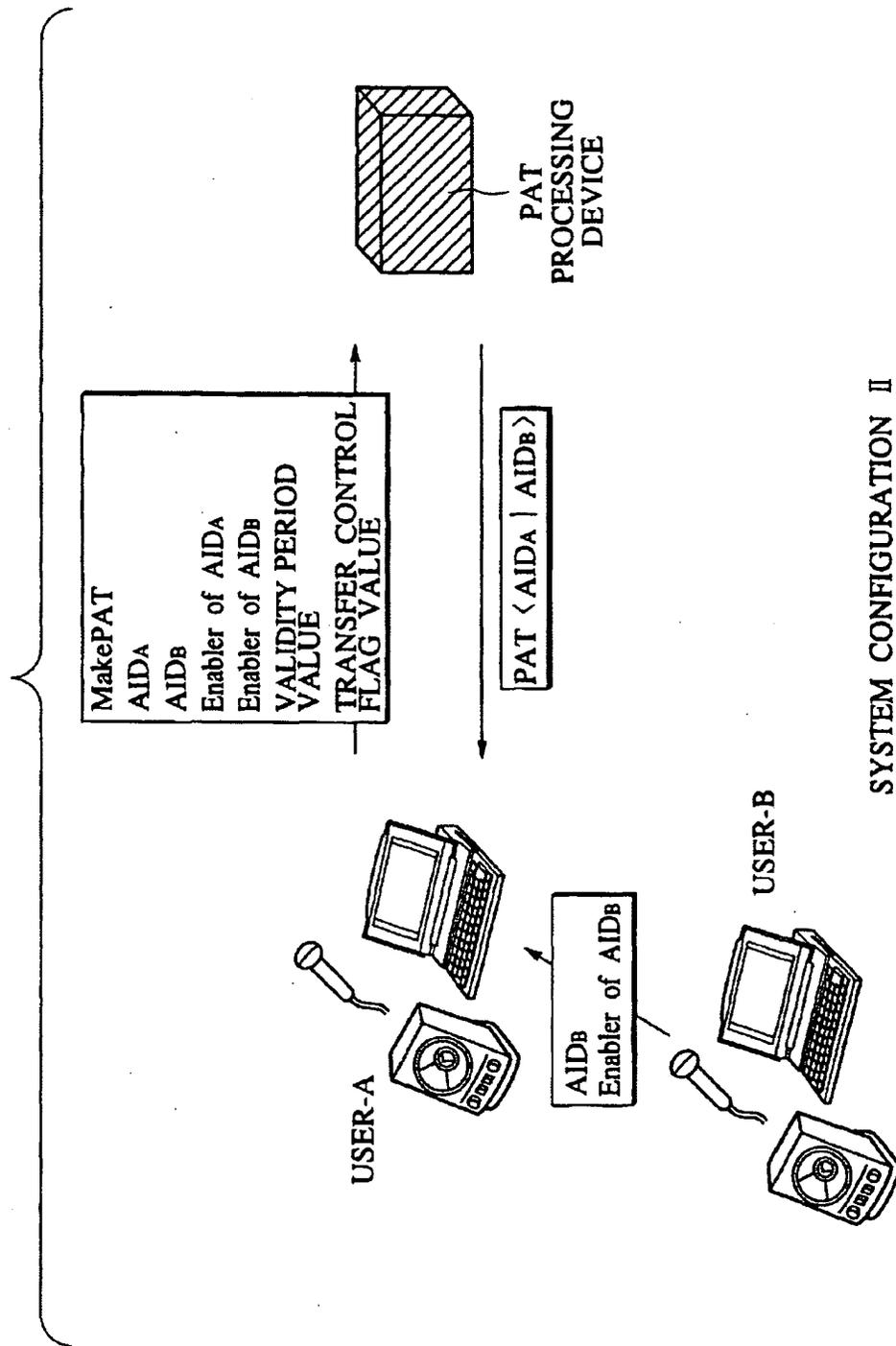
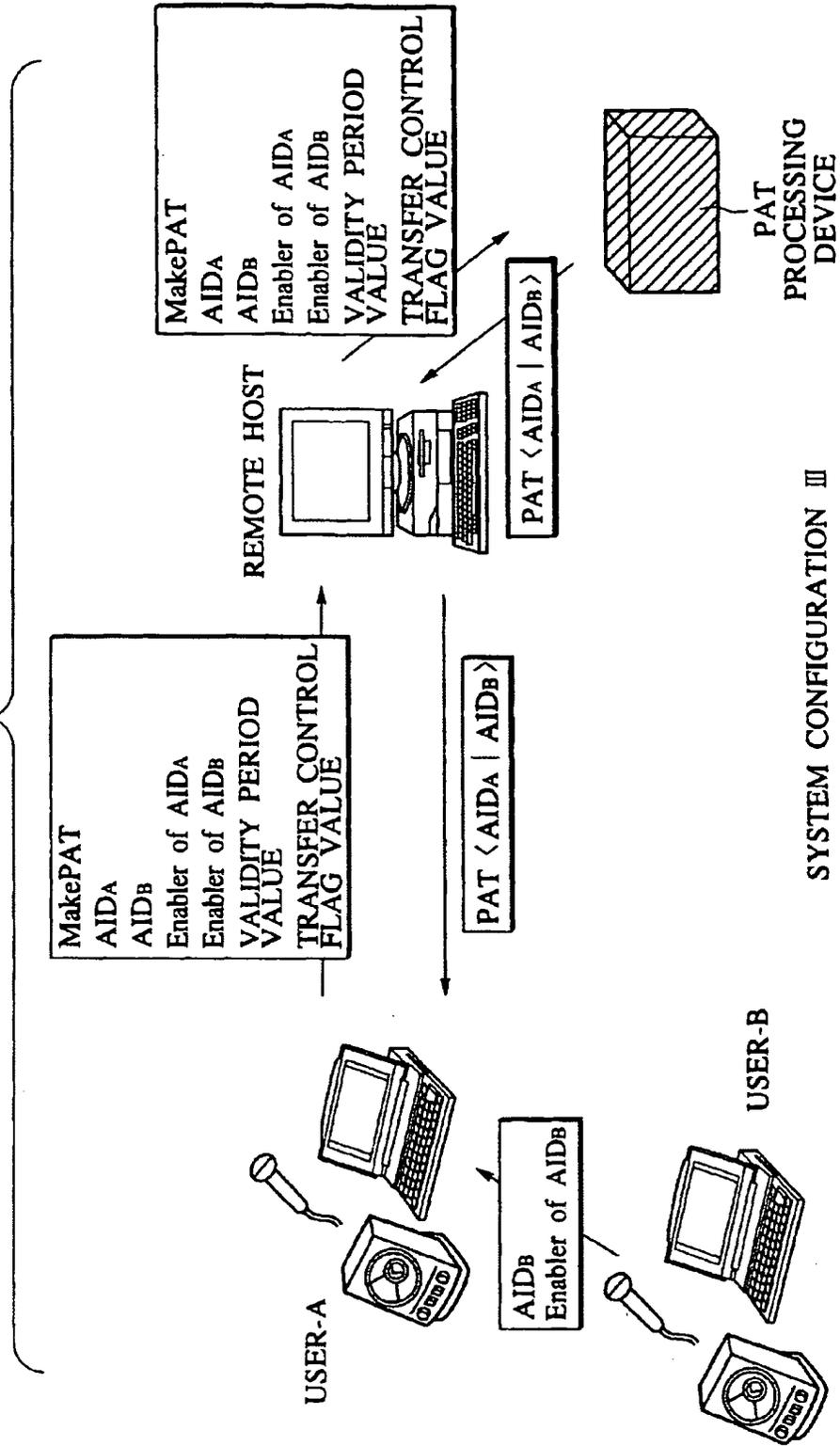
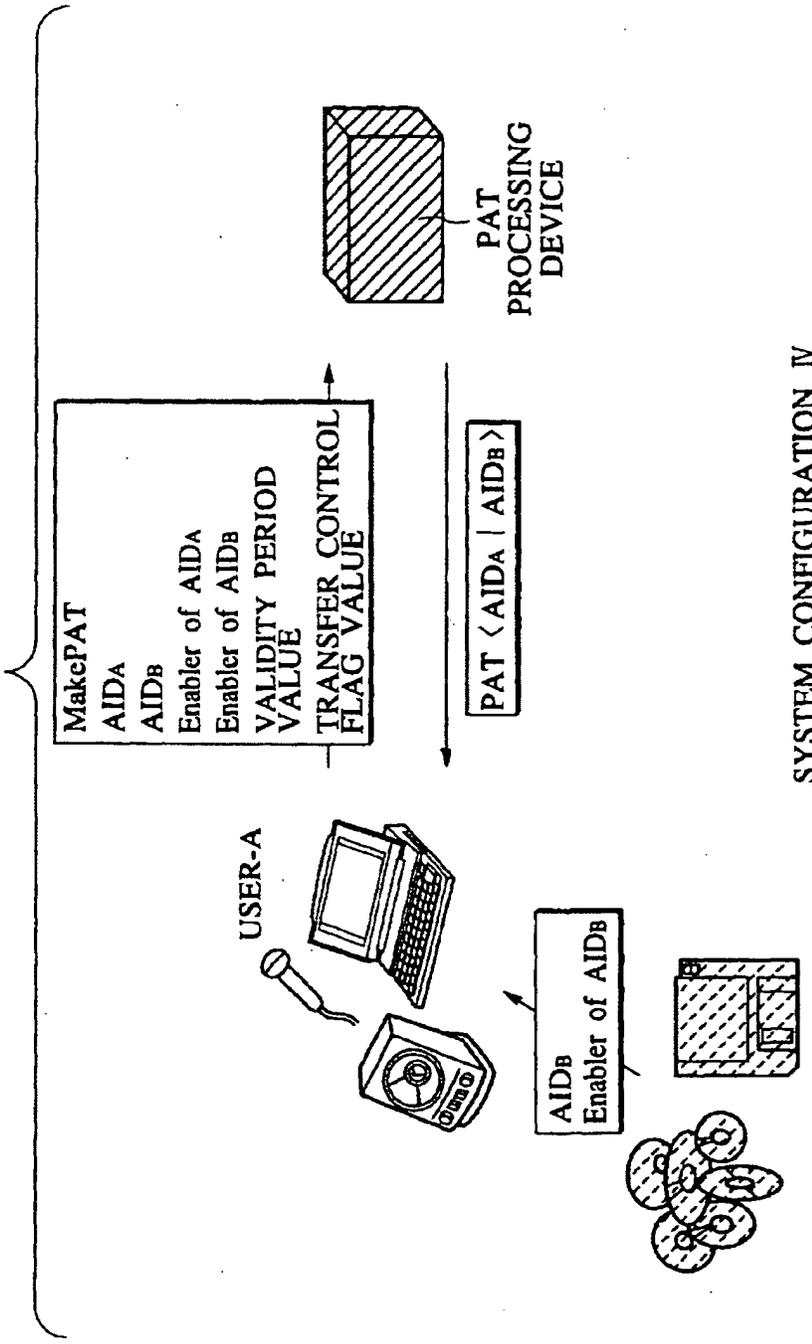


FIG.16



SYSTEM CONFIGURATION III

FIG.17



SYSTEM CONFIGURATION IV

FIG. 18

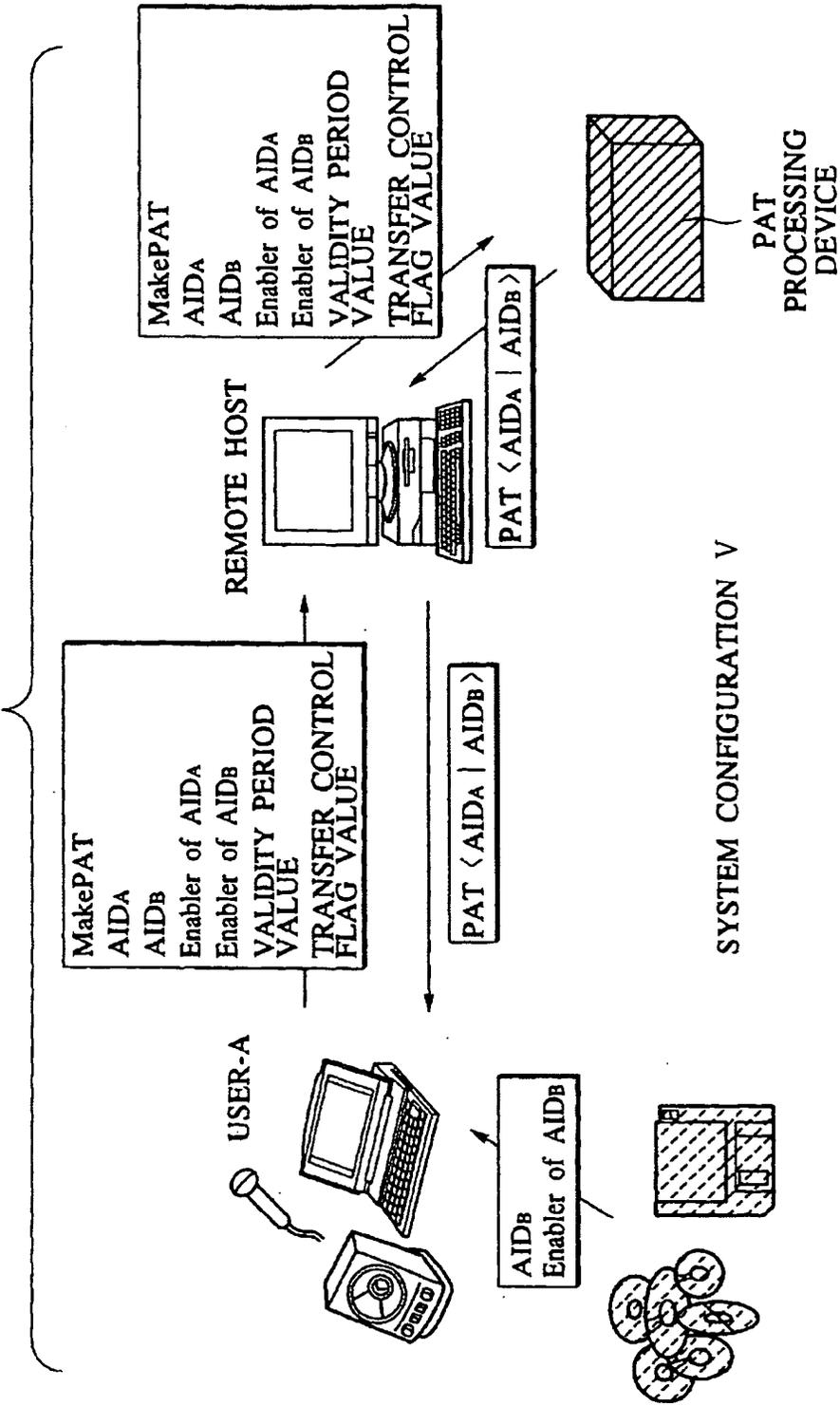


FIG. 19

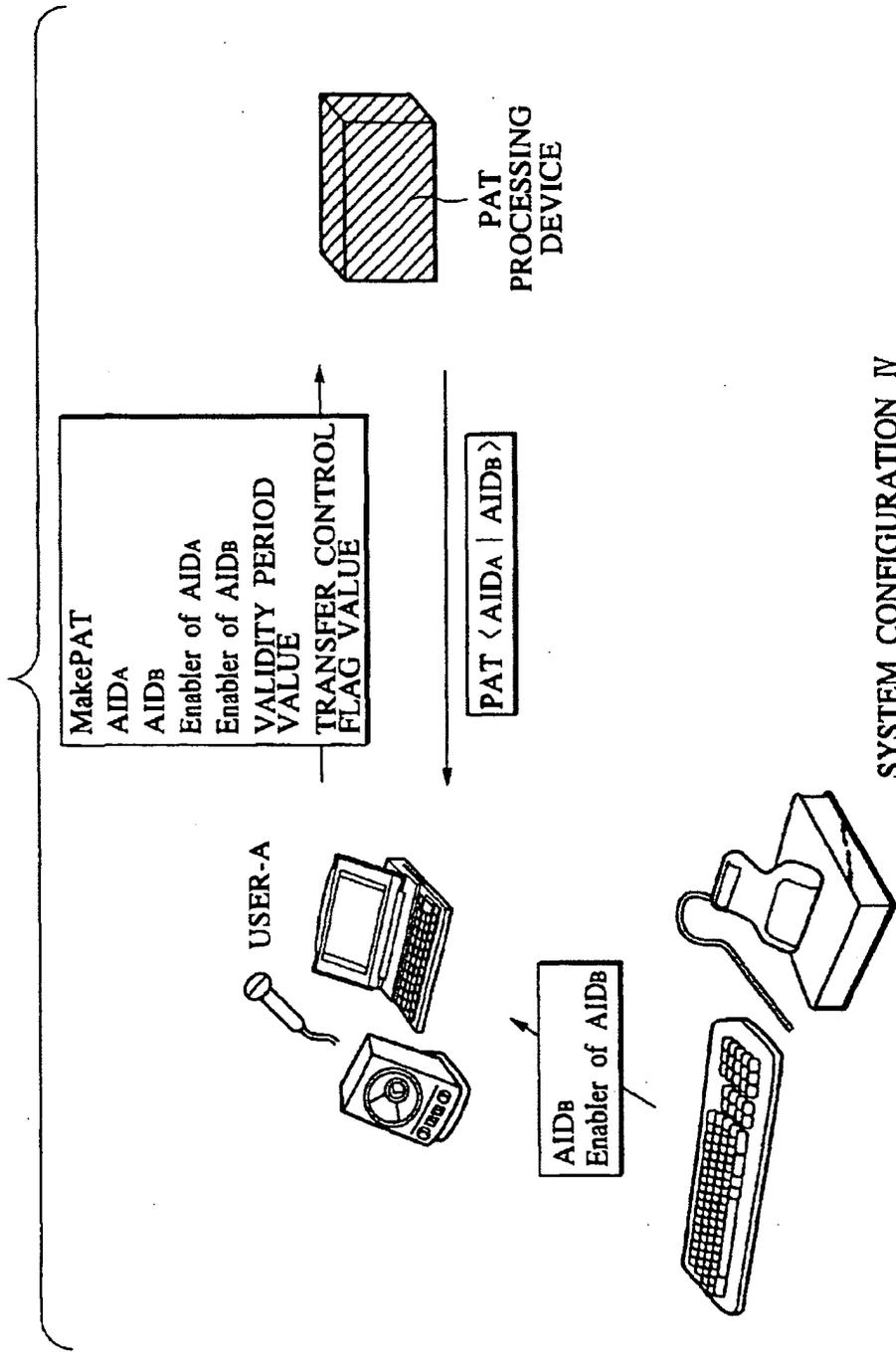


FIG. 20

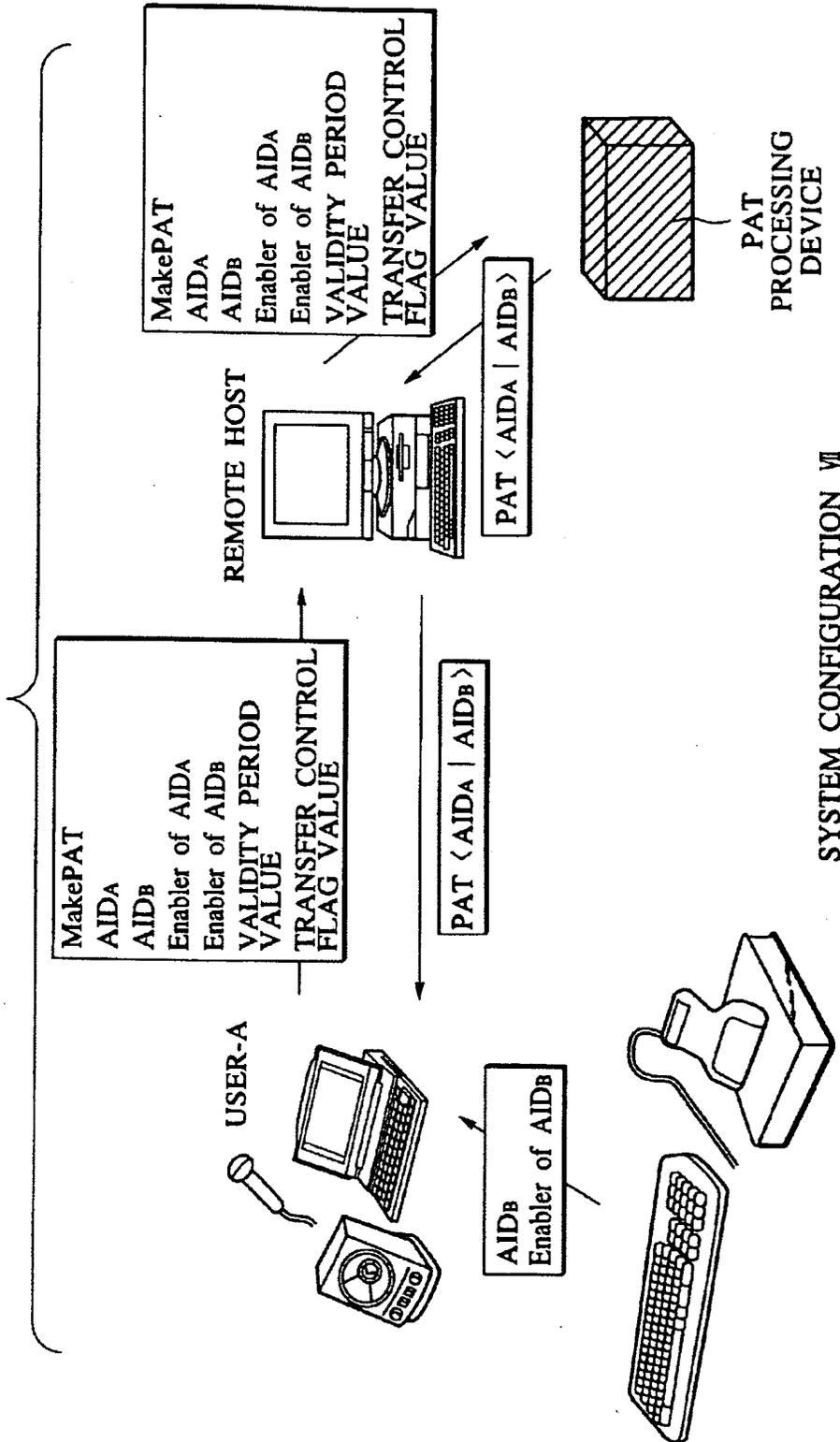


FIG.21

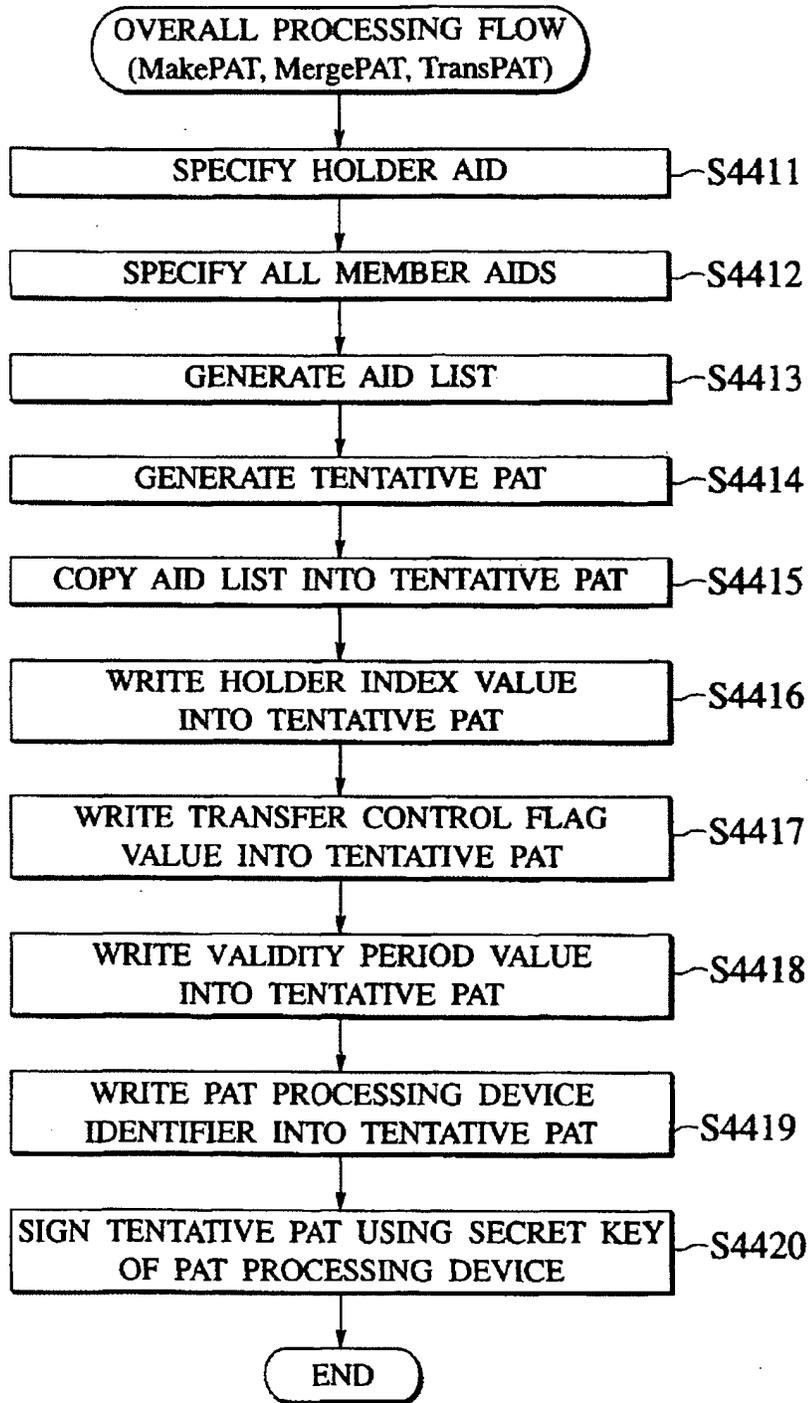


FIG.22

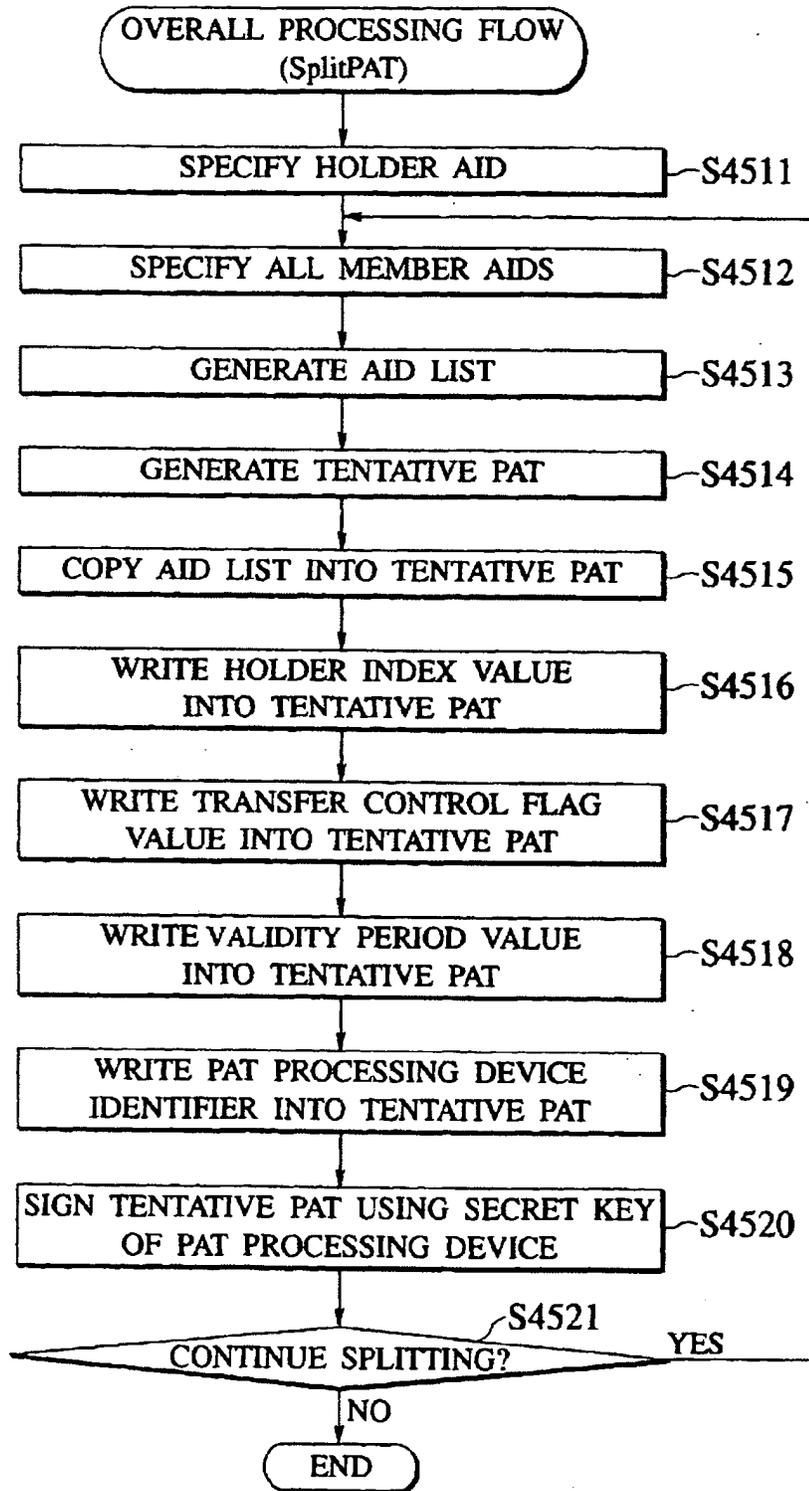


FIG.23

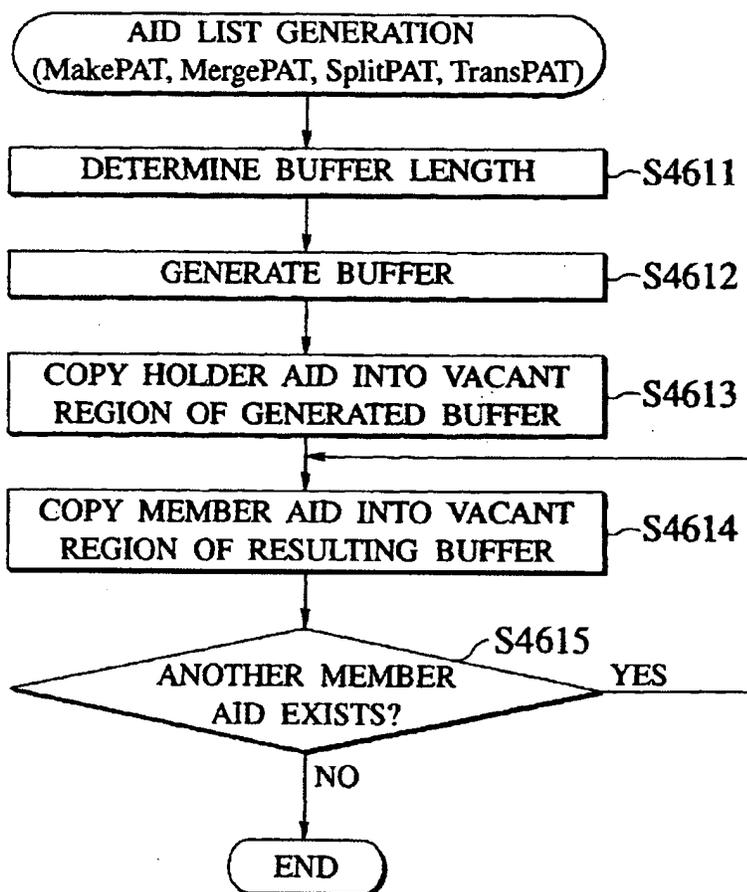


FIG.24

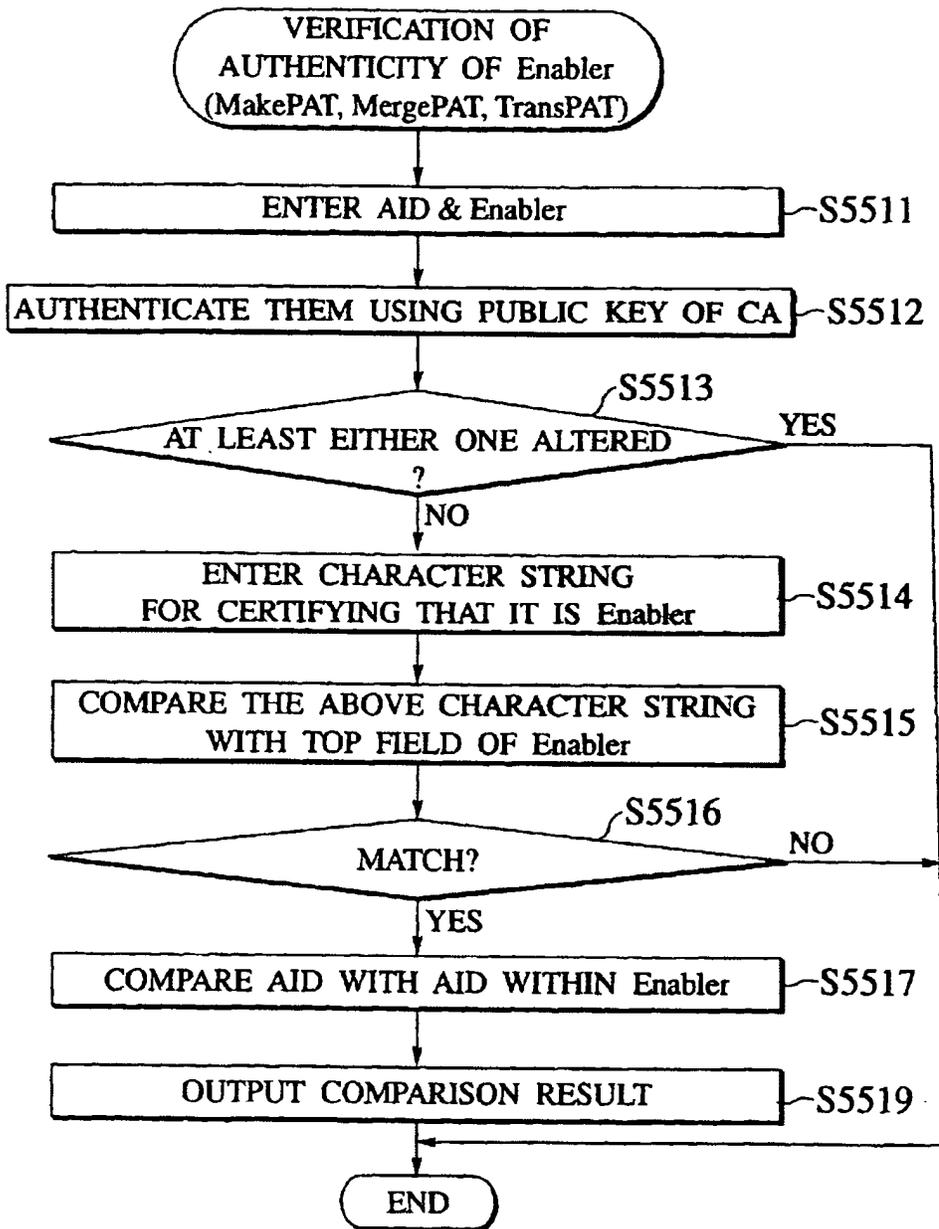


FIG.25

DATA STRUCTURE OF Null-AID

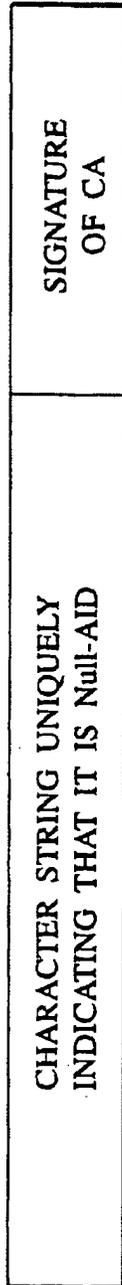


FIG.26

DATA STRUCTURE OF Enabler of Null-AID

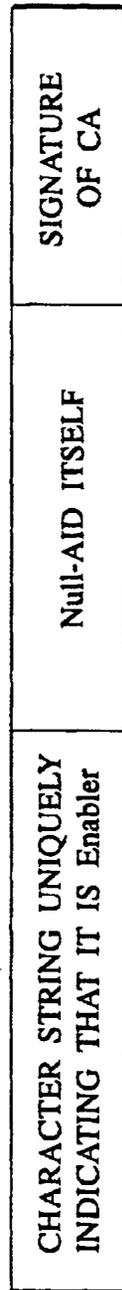


FIG.27

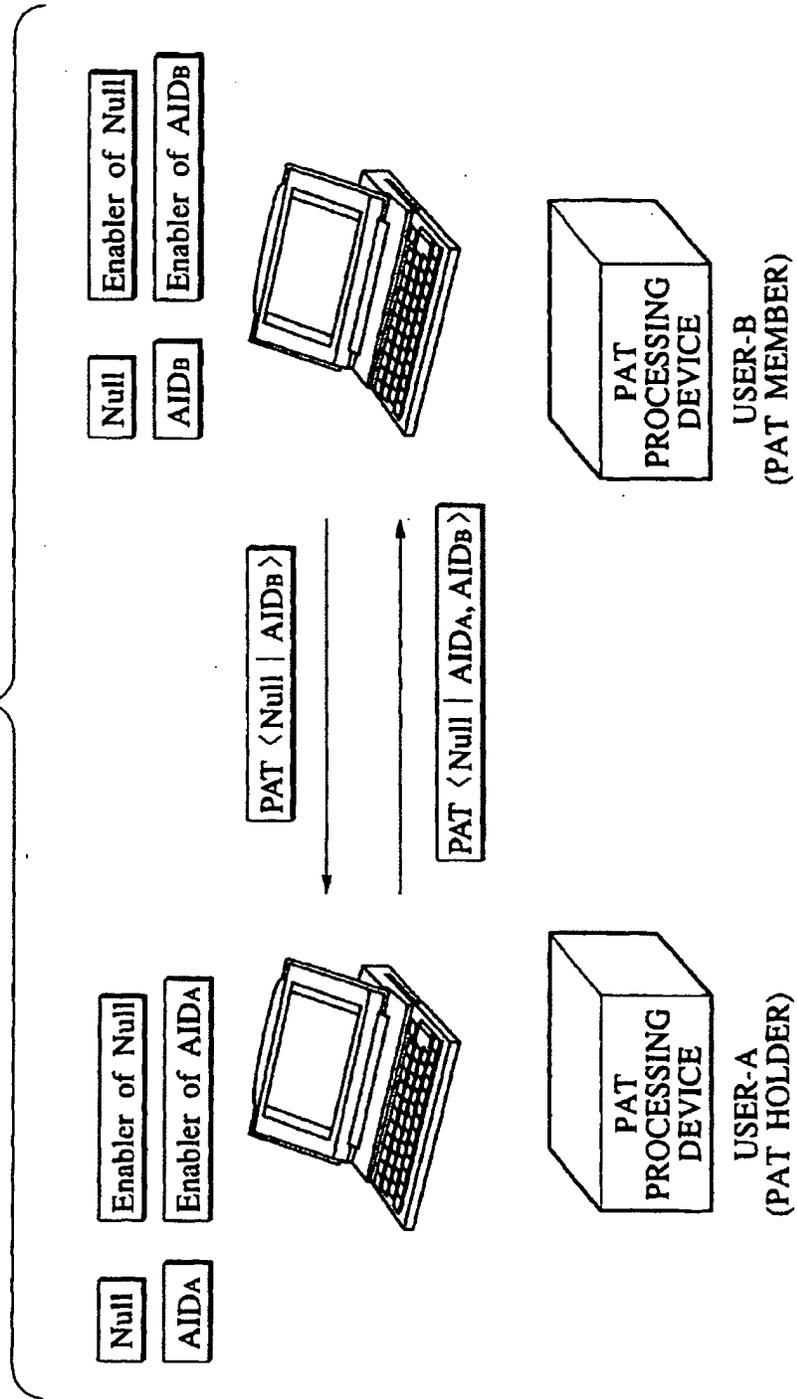


FIG.28

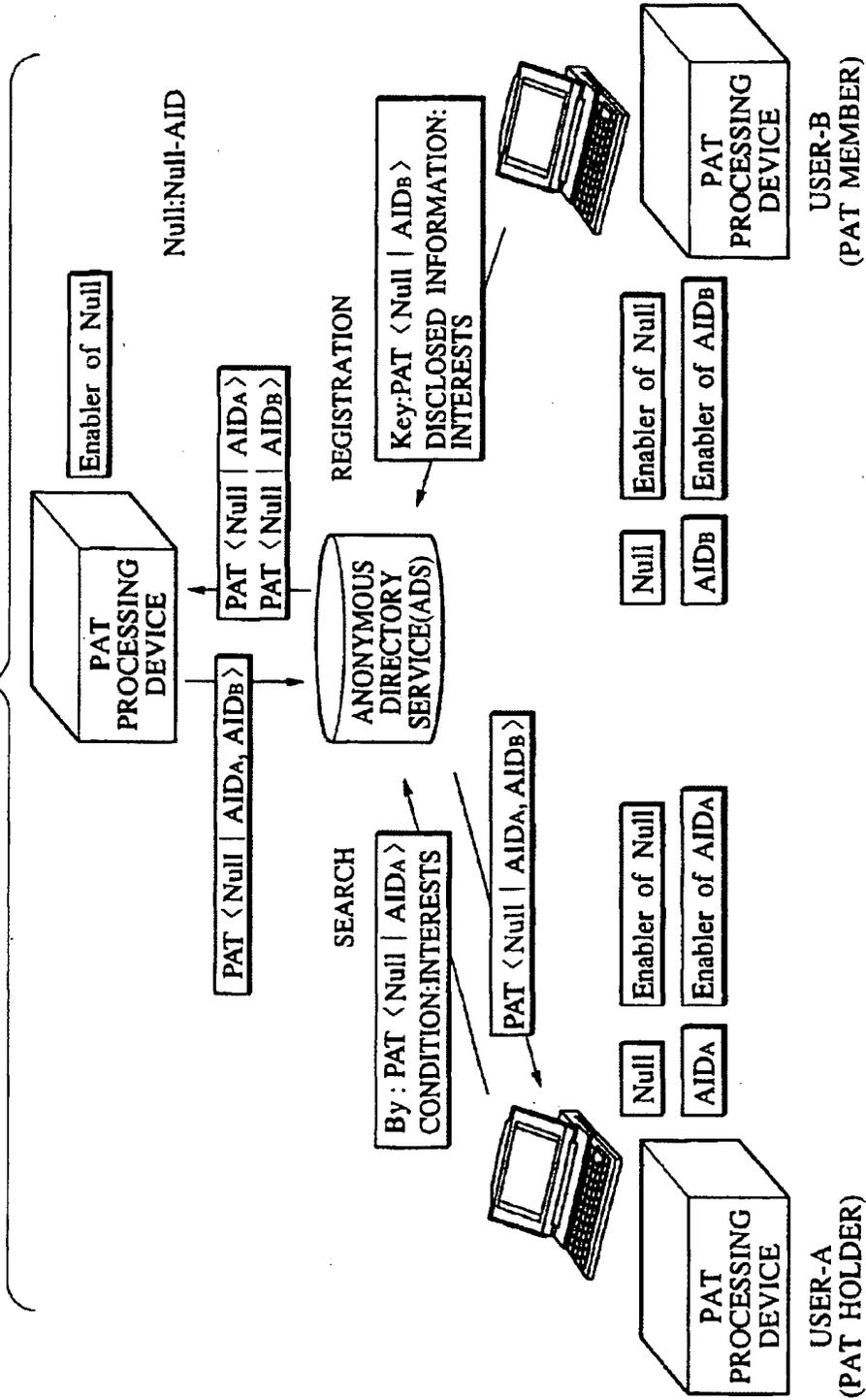


FIG.29

DATA STRUCTURE OF God-AID

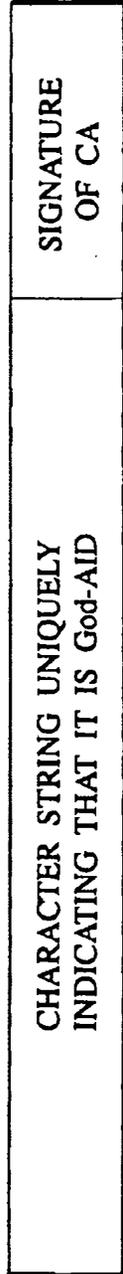


FIG.30

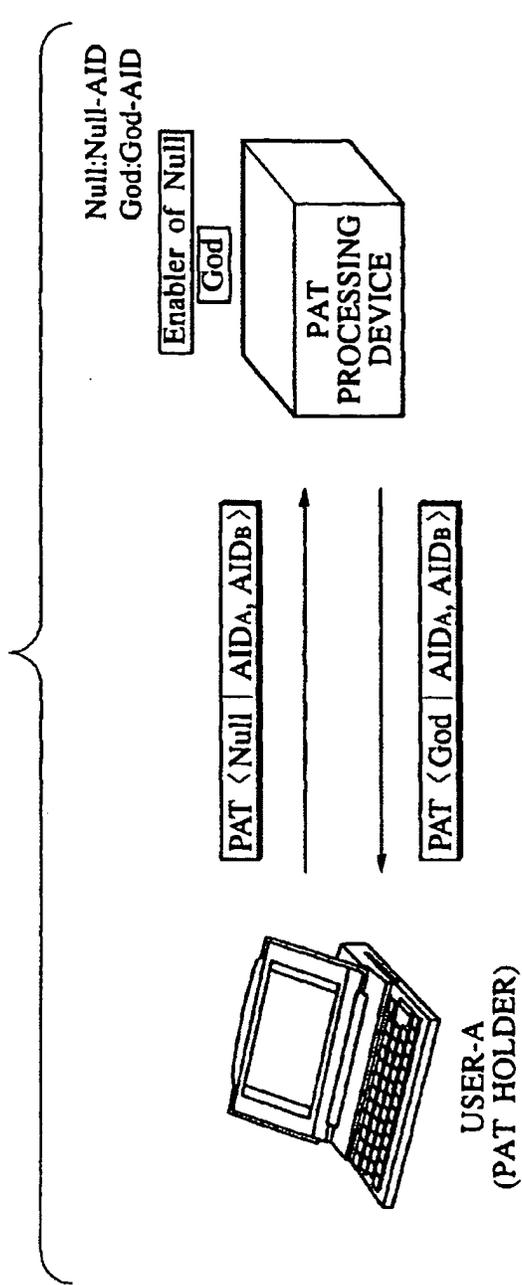


FIG.31

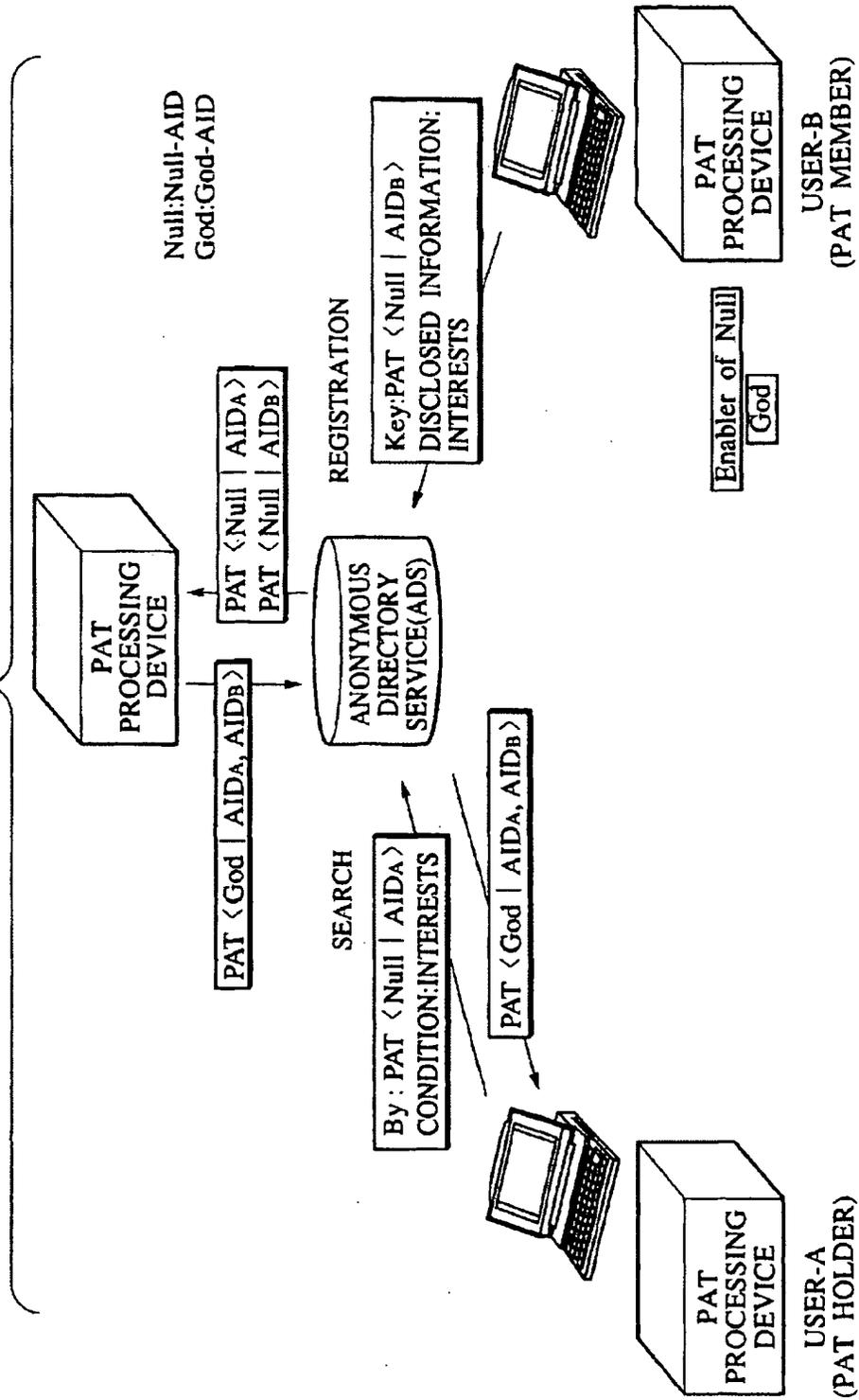


FIG.32

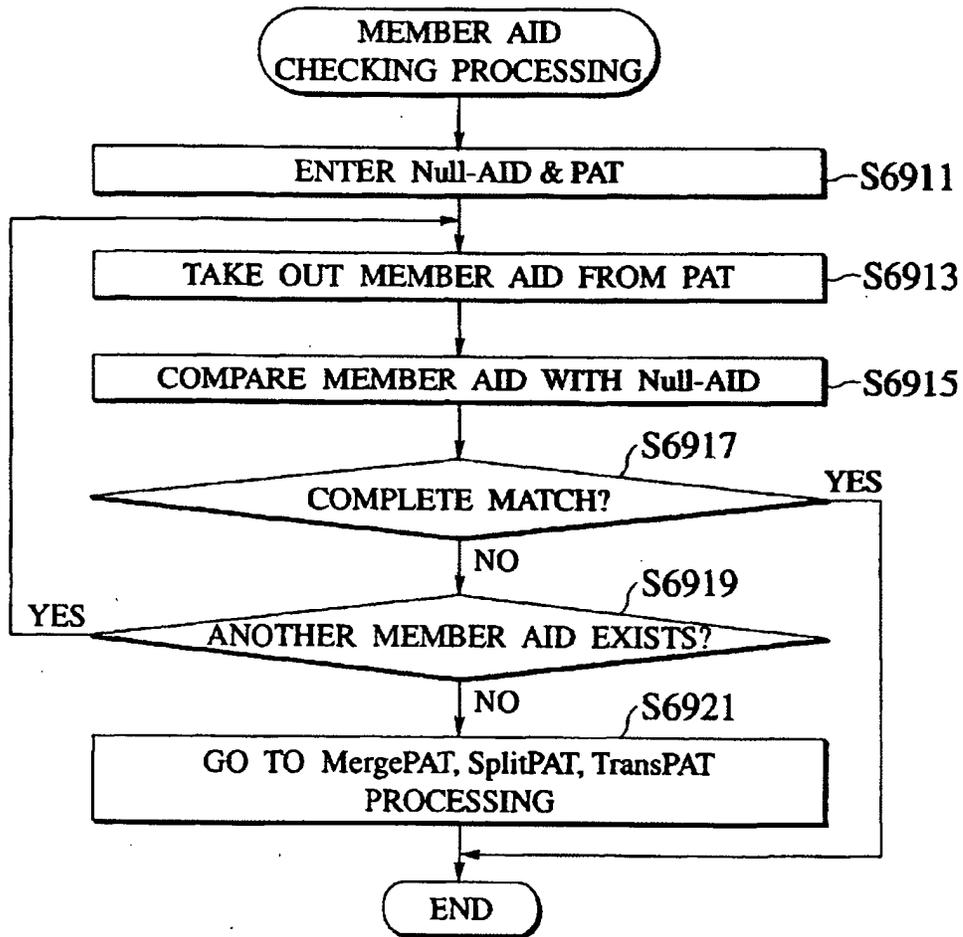


FIG.33

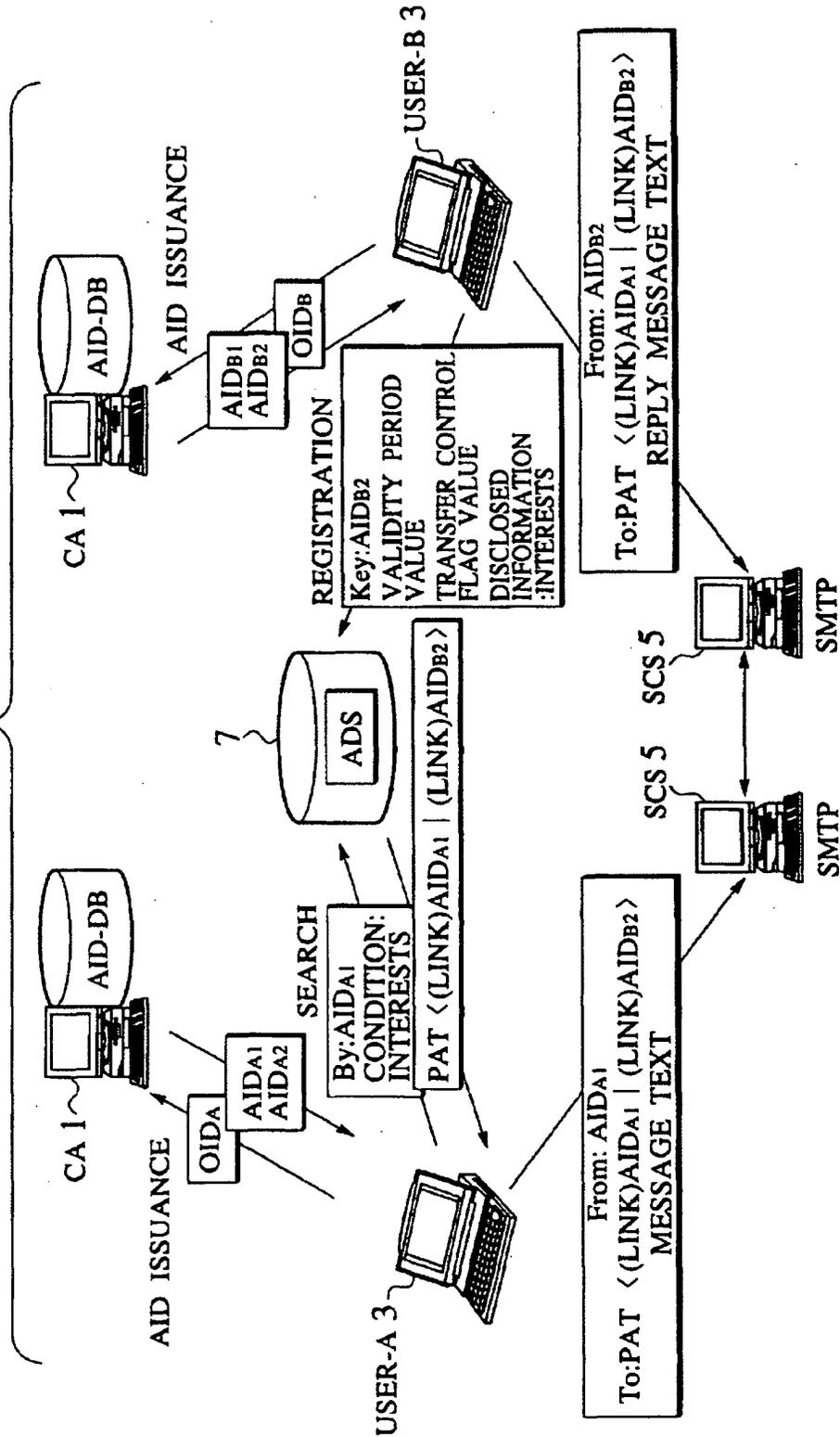


FIG.34

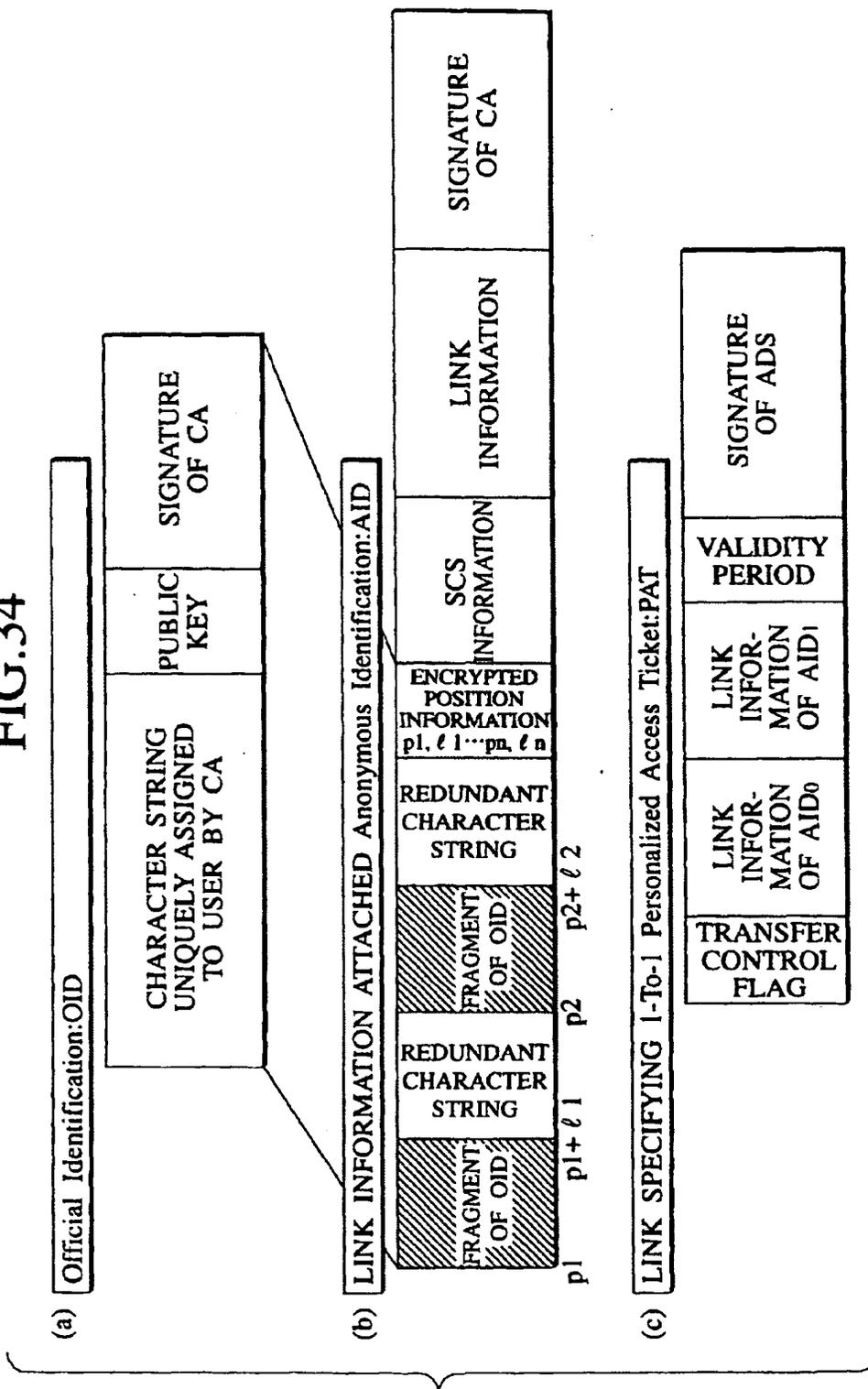


FIG.35

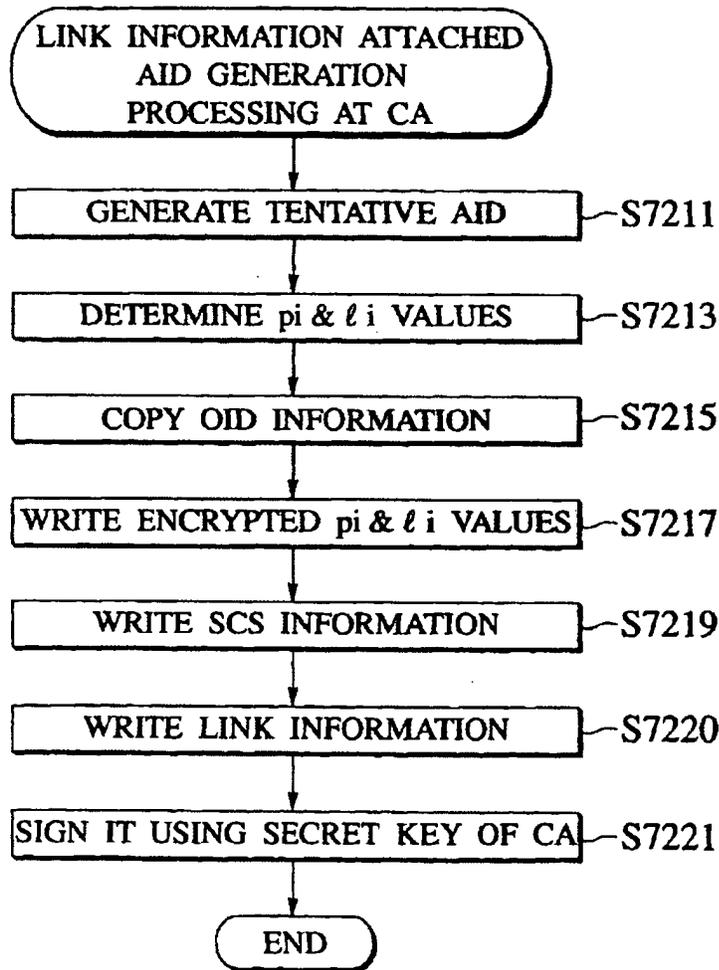


FIG.36

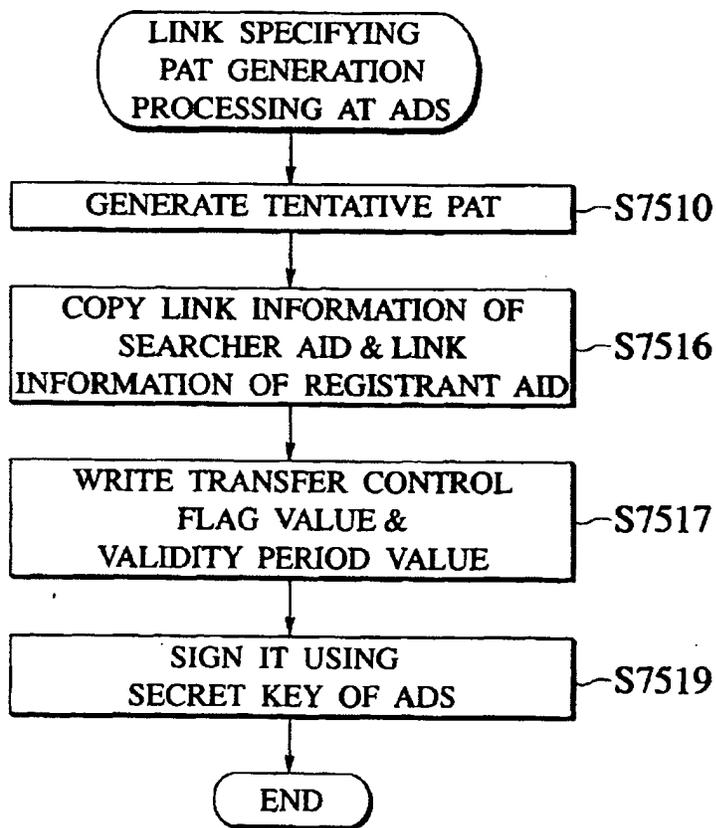


FIG.37

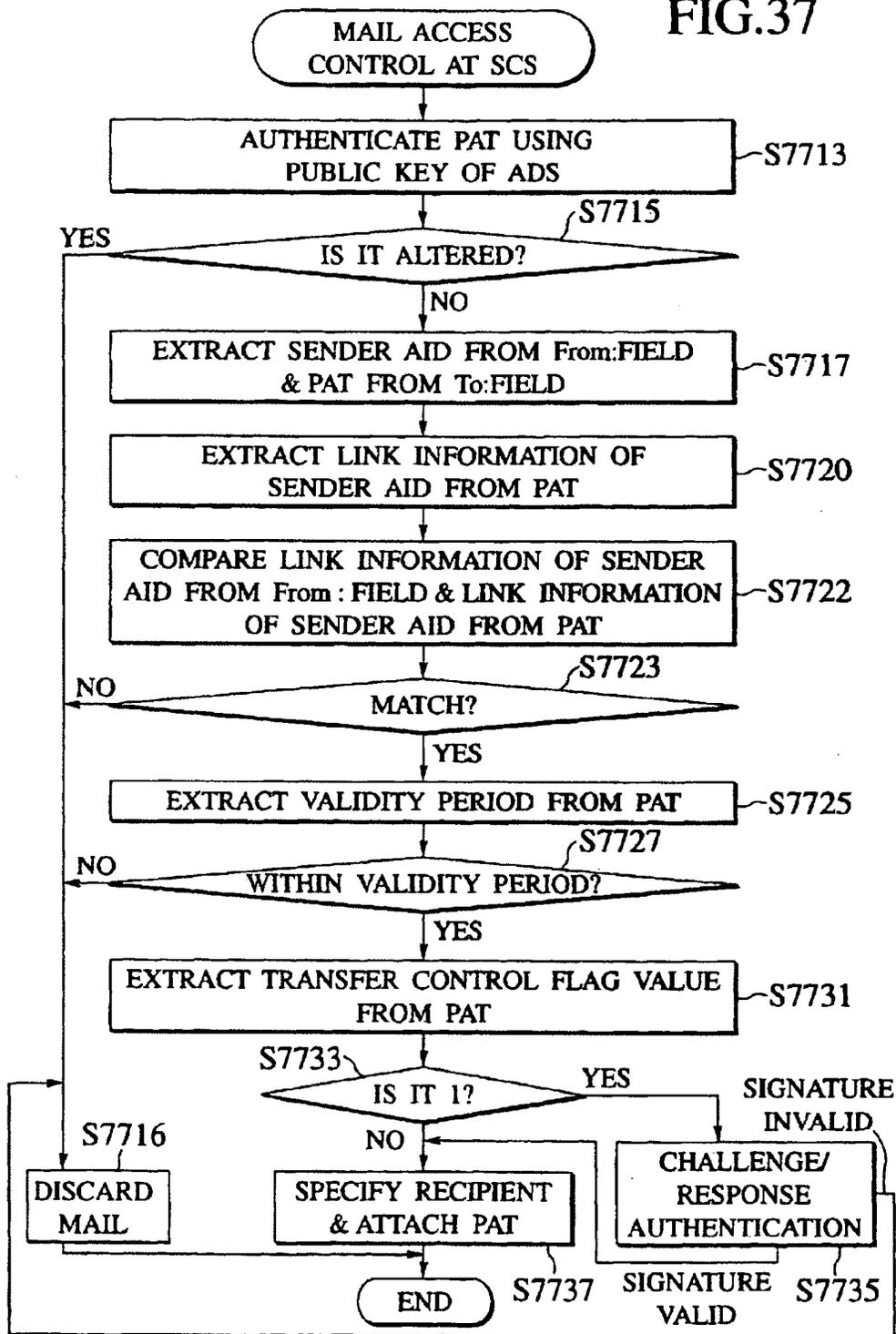


FIG.38

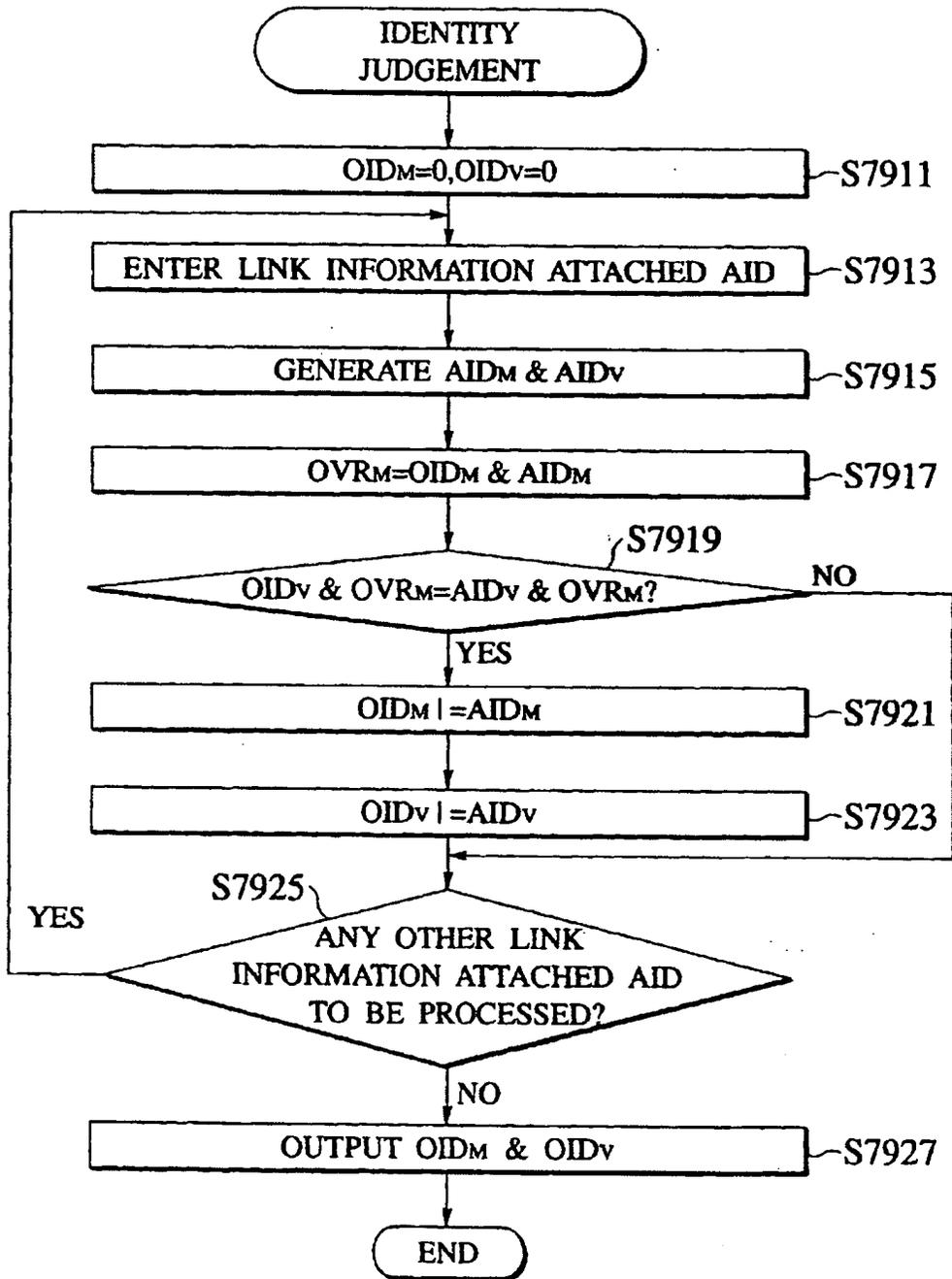


FIG.39

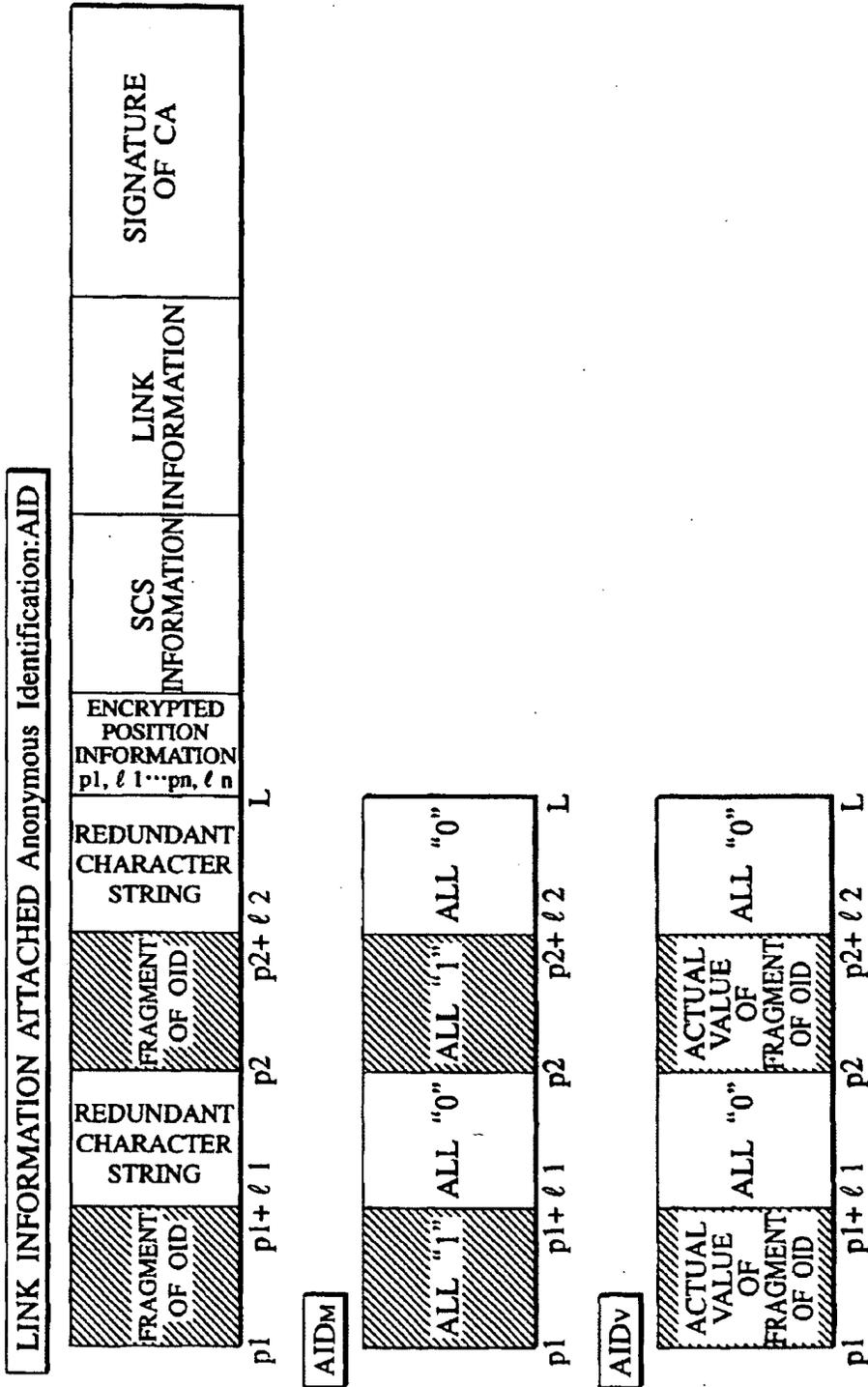
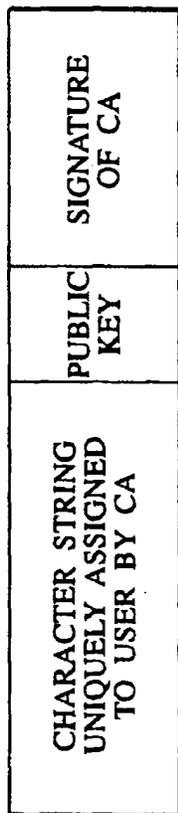
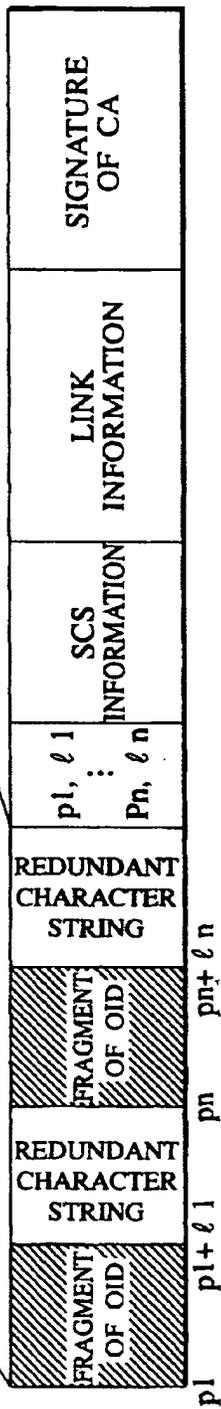


FIG.40

(a) Official Identification:OID



(b) LINK INFORMATION ATTACHED Anonymous Identification:AID



(c) LINK SPECIFYING 1-To-N Personalized Access Ticket:PAT

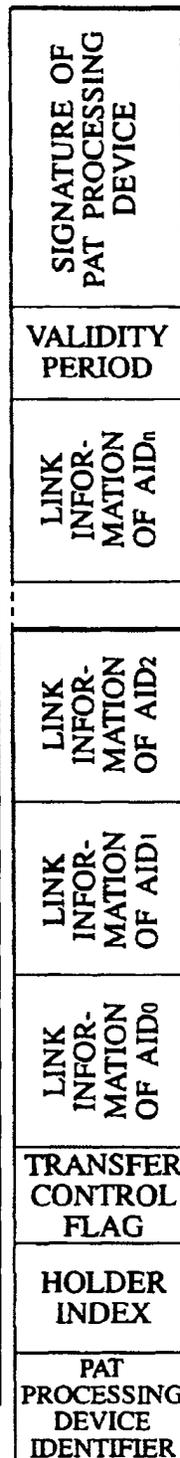


FIG.41

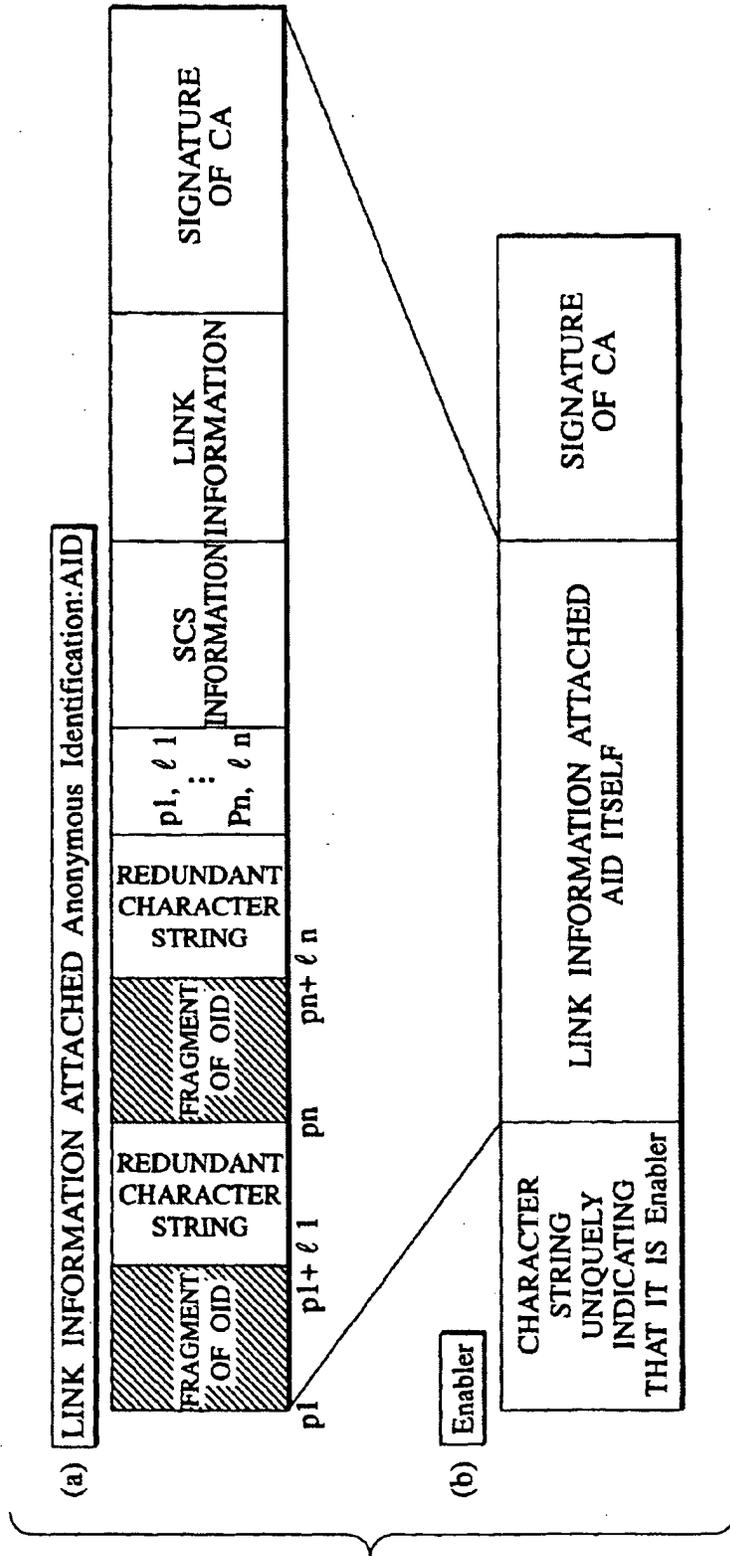


FIG.42

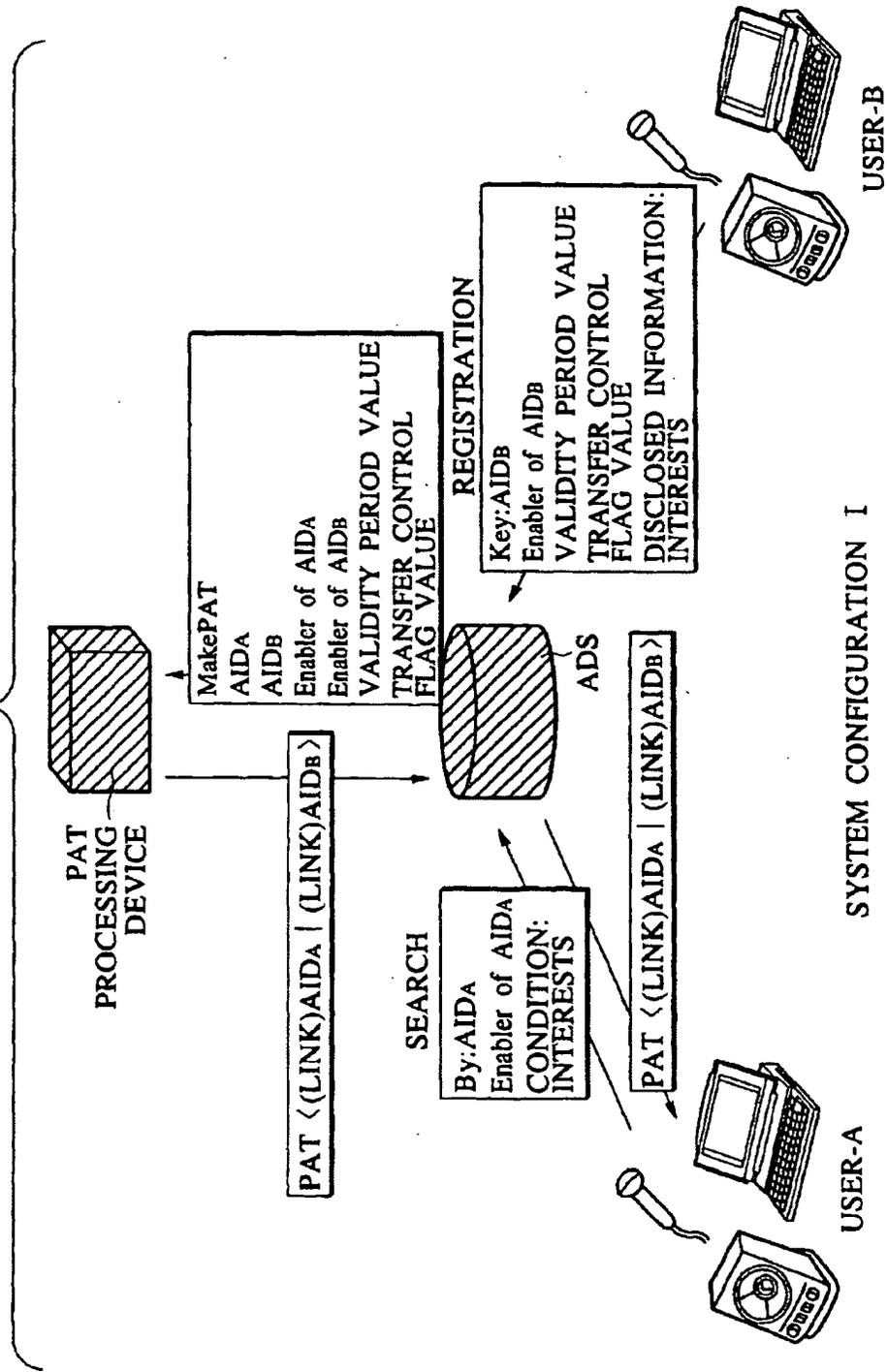
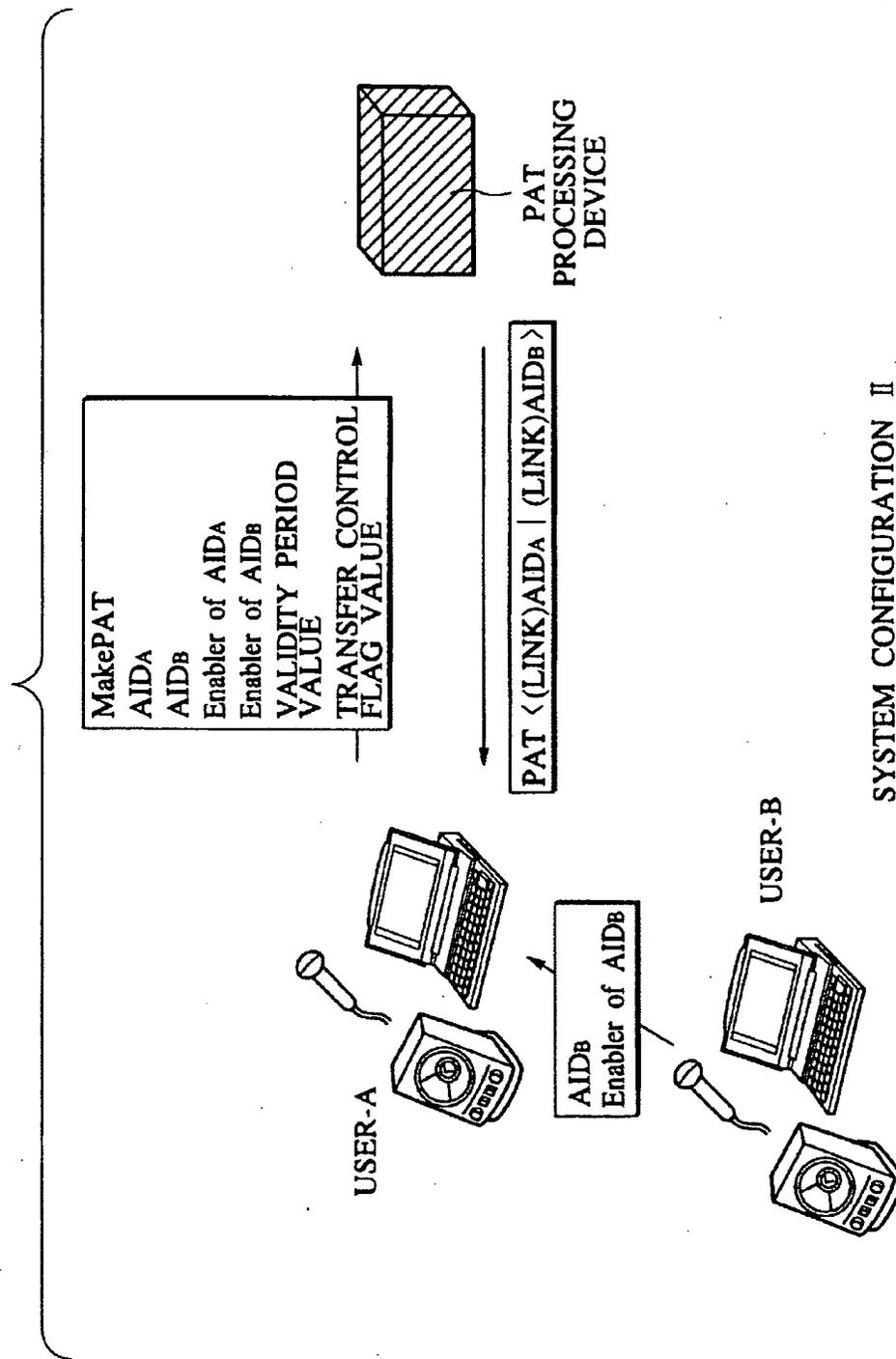


FIG.43



SYSTEM CONFIGURATION II

FIG.44

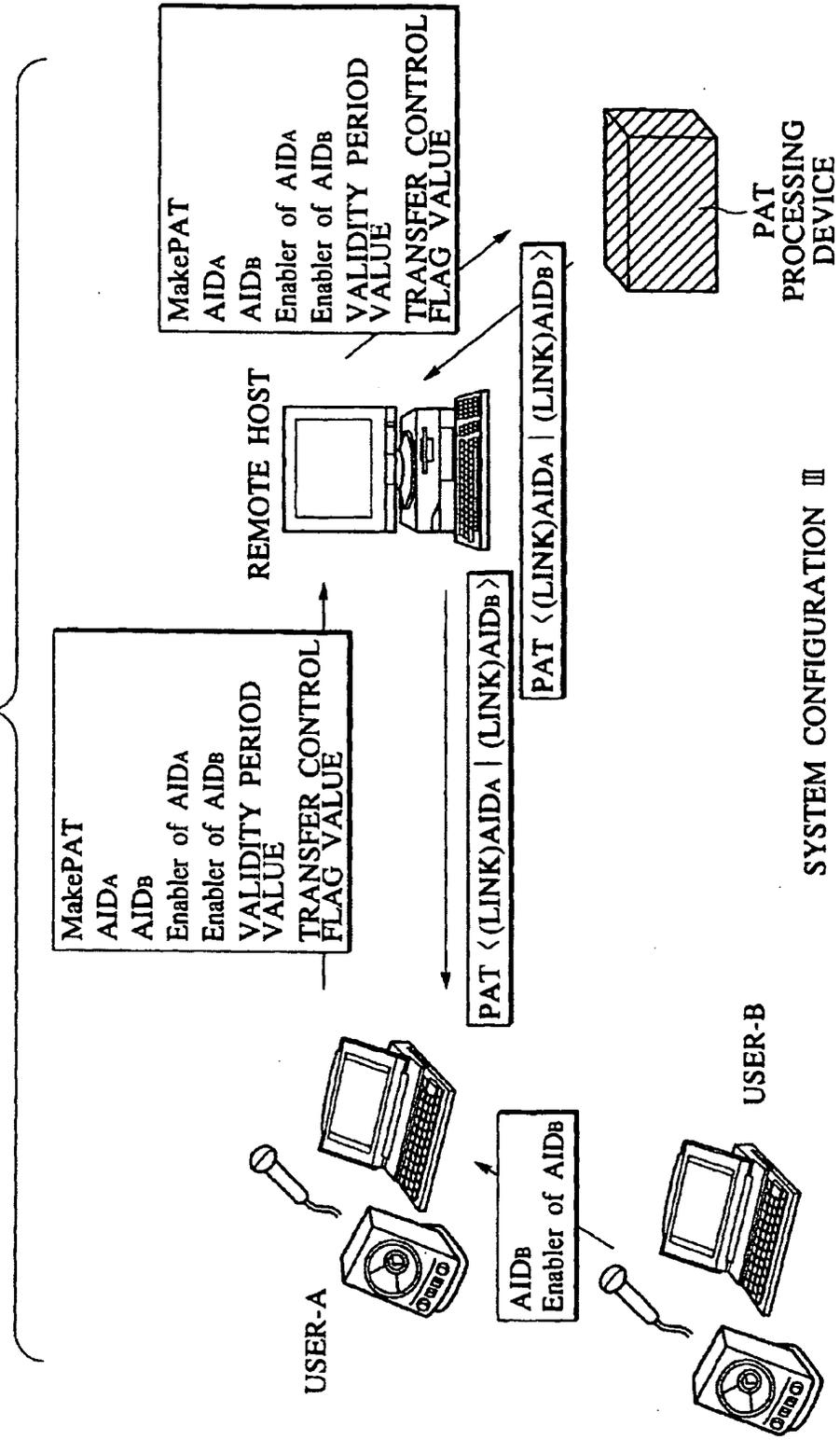


FIG.45

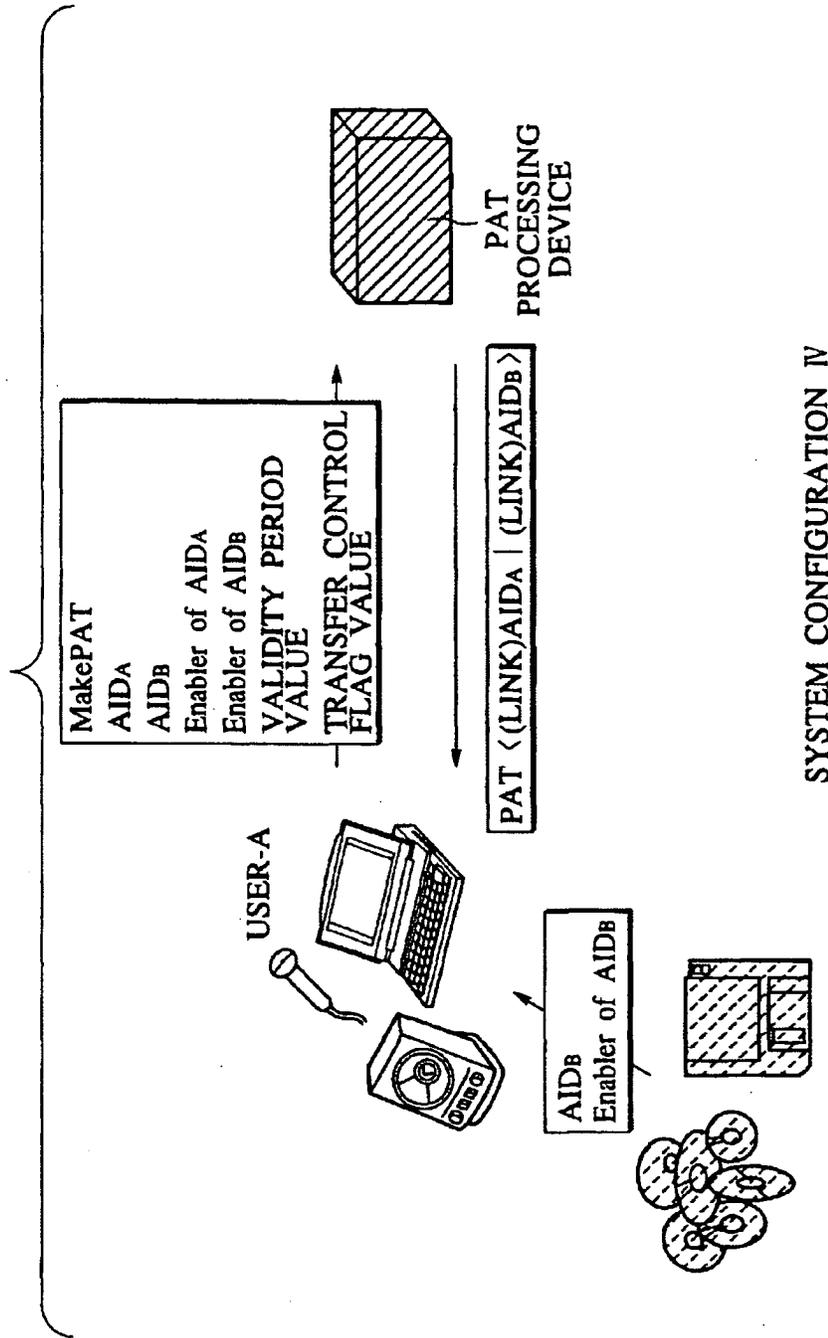


FIG.46

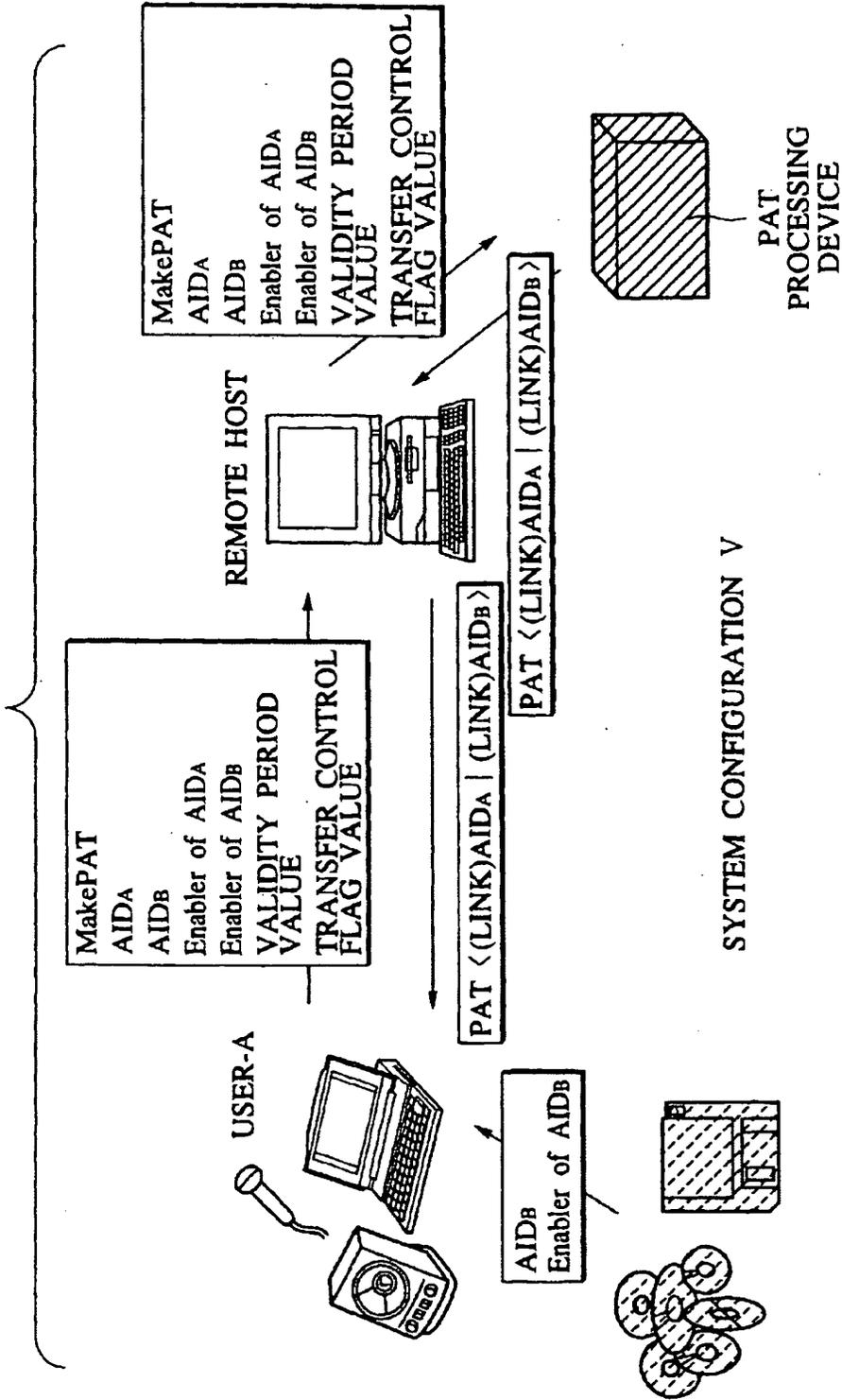


FIG.47

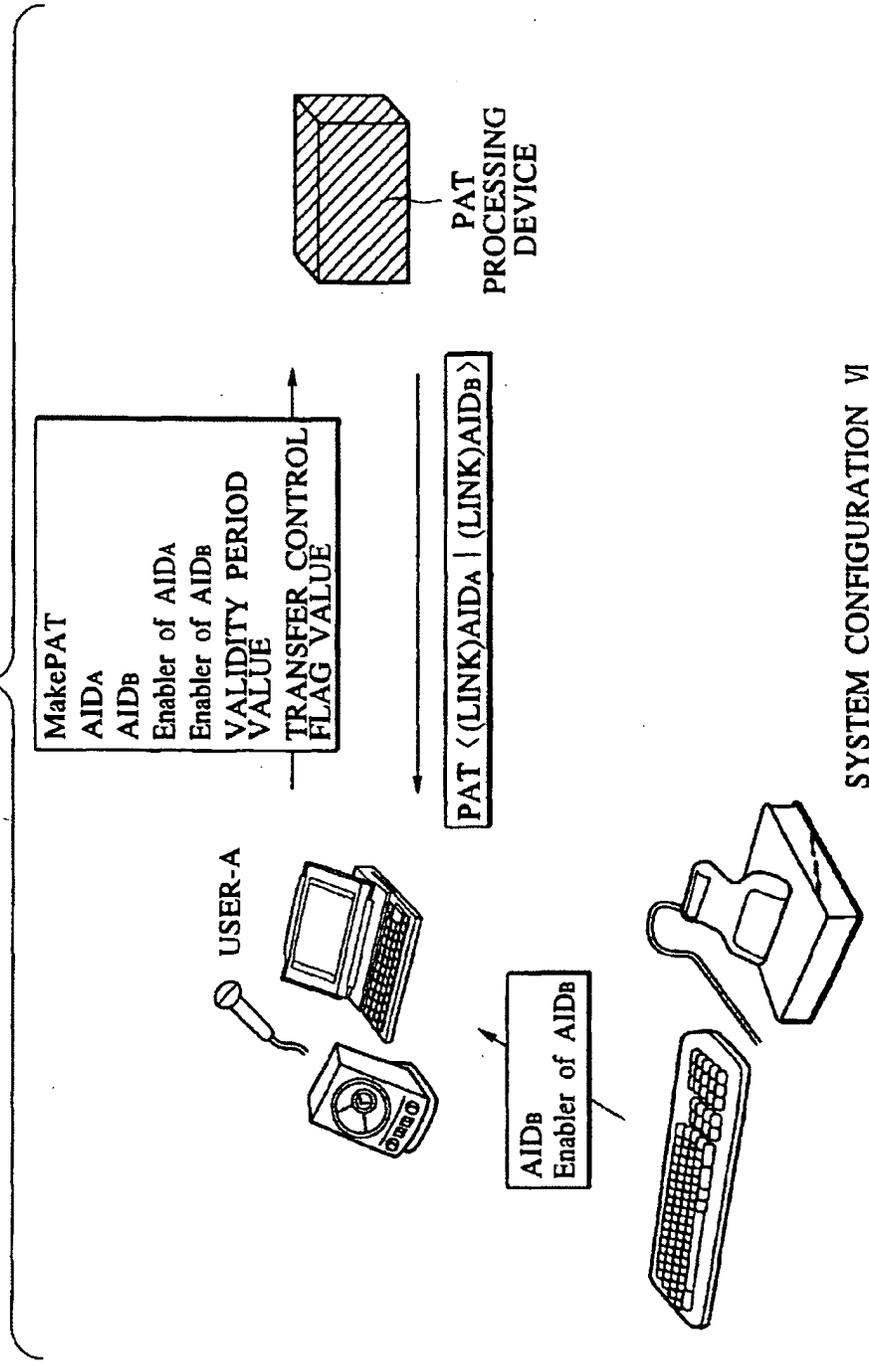
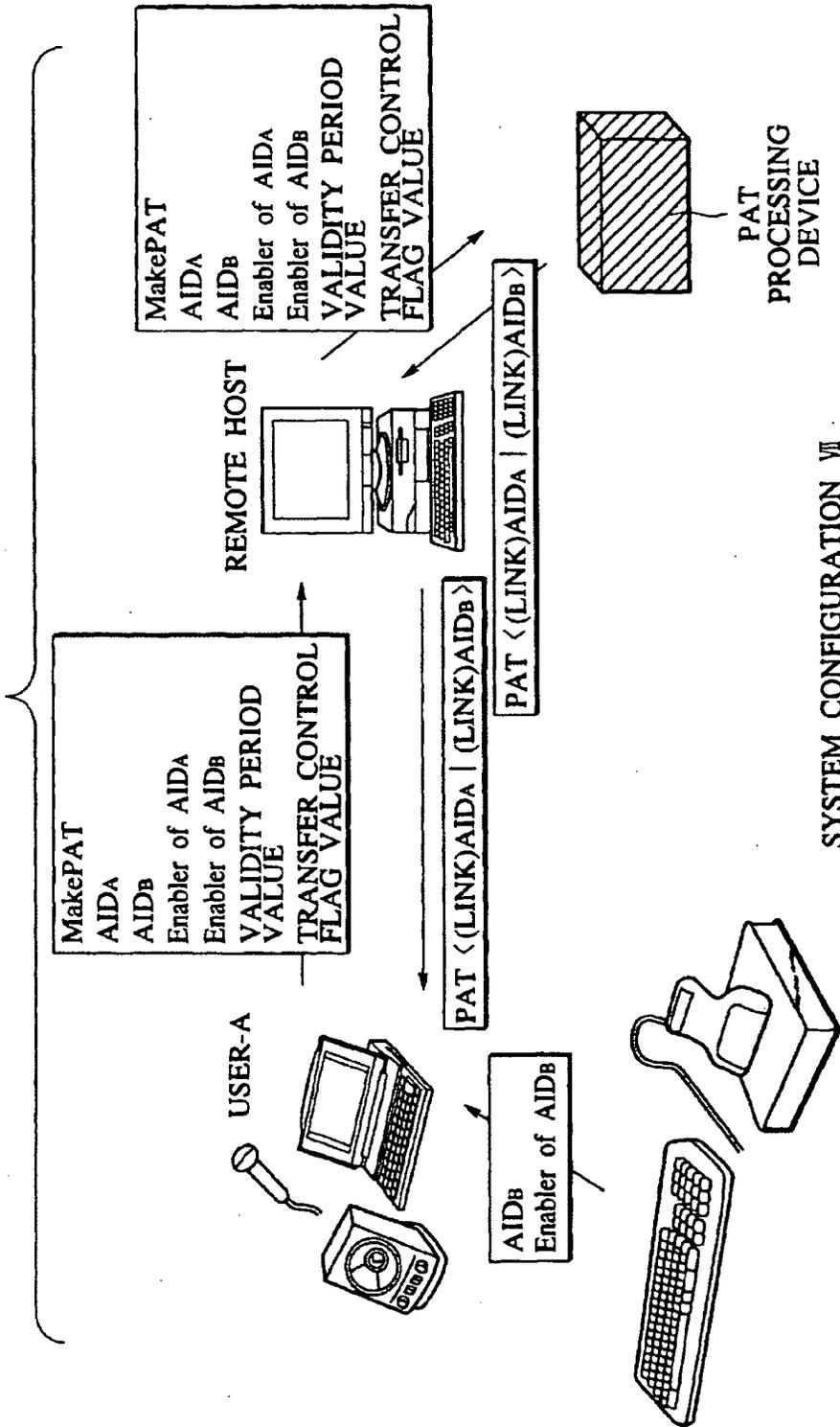
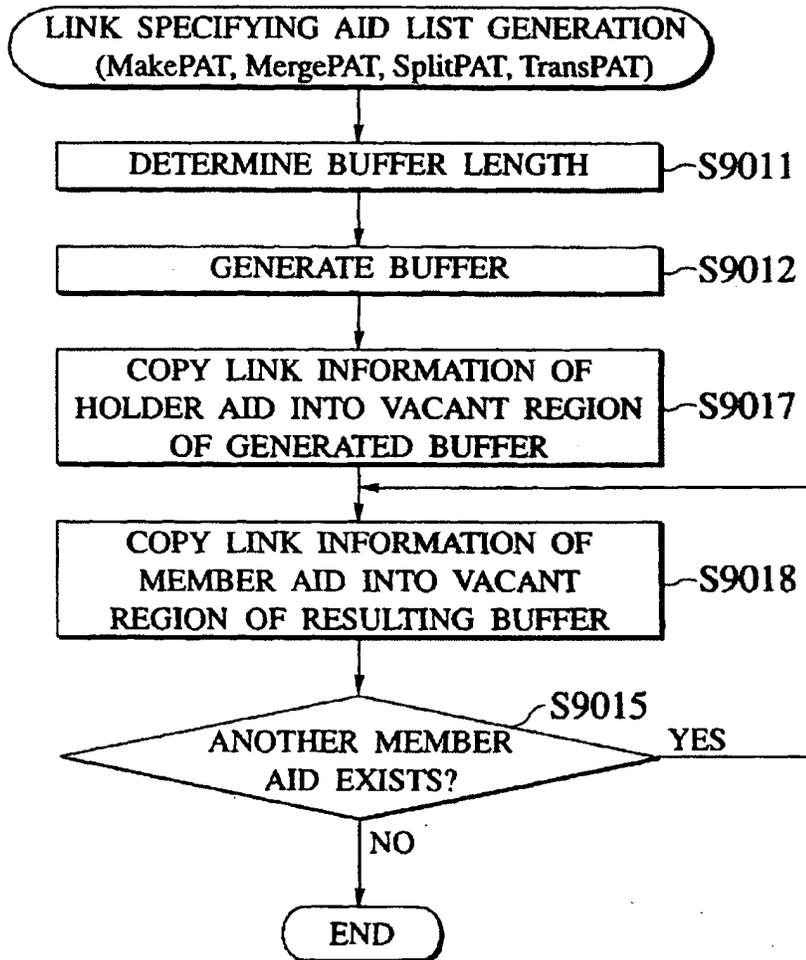


FIG. 48



SYSTEM CONFIGURATION VII

FIG.49





(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication: 15.12.1999 Bulletin 1999/50
 (51) Int. Cl.⁶: H04N 5/00
 (21) Application number: 98401374.8
 (22) Date of filing: 08.06.1998

<p>(84) Designated Contracting States: AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE Designated Extension States: AL LT LV MK RO SI</p> <p>(71) Applicant: CANAL+ Société Anonyme 75711 Paris Cedex 15 (FR)</p>	<p>(72) Inventor: Declerck, Christophe 28210 Senantes (FR)</p> <p>(74) Representative: Cozens, Paul Dennis et al Mathys & Squire 100 Grays Inn Road London WC1X 8AL (GB)</p>
---	---

(54) **Decoder and security module for a digital transmission system**

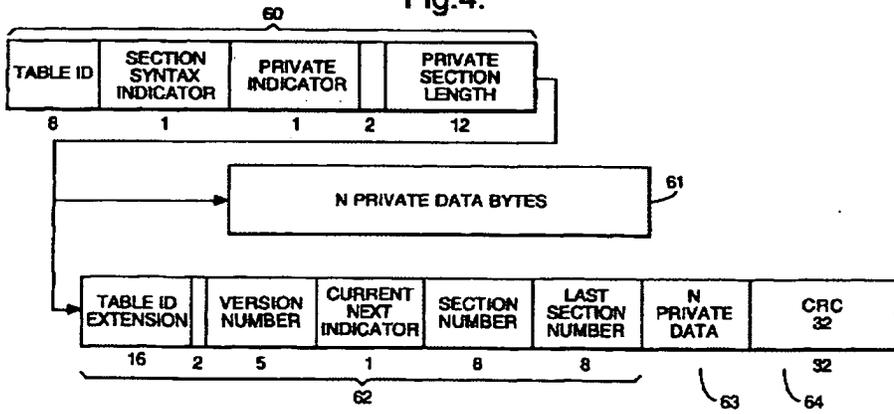
(57) A decoder 12 in particular for a digital television system and adapted to receive a transport packet stream containing table or section data encapsulated within the packet payloads. The decoder is characterised in comprising a means 80 for filtering table or section data configurable in response to filter data received from a portable security module 30 such as a smart card.

necessary to configure the table or section filter 80, and a method for processing a transport packet stream including encapsulated table and section data using such a decoder 12 and security module 30.

In a preferred embodiment, the filter 80 is adapted to filter out conditional access messages in response to the table or section filter data received from the portable security module 30, these messages being thereafter forwarded to the security module for processing.

The invention equally extends to a portable security module 30 including a memory holding such data as is

Fig.4.



EP 0 964 572 A1

Description

[0001] The present invention relates to a decoder and security module for a digital transmission system and method of operating a decoder and security module, in particular for use in a digital television system.

5 [0002] Conventional digital television broadcast systems transmit data in the form of discrete transport stream packets or transport packets, each packet being of a predetermined length and containing a header and a payload. The MPEG standard is the currently favoured standard in this domain and sets out, amongst other things, a predetermined format for such packets.

10 [0003] The packet header comprises general descriptive data regarding the packet, whilst the payload comprises the data to be processed at the receiver. The packet header includes at least a packet ID or PID identifying the packet. The payload of the packet may contain audio, video or other data such as application data or, in particular, conditional access system data.

15 [0004] Conventionally, the incoming data stream is filtered by a receiver/decoder according to the PID of each packet. Data requiring immediate processing such as audio or visual data is communicated to an appropriate processor in the form of what is conventionally known as a packetised elementary stream or PES. This continuous flux of data, which is formed by assembling the payloads of the transport packets, itself comprises a sequence of packets, each PES packet comprising a packet header and payload.

20 [0005] Other data not requiring immediate processing may also be encapsulated within the payloads of the transport packets. Unlike PES data, which is treated immediately by a processor to generate a real time output, this sort of data is typically processed in an asynchronous manner by the decoder processor. In this case, data is formatted in a single table or a series of sections or tables, each including a header and a payload, the header of the section or table including a table ID or TID.

25 [0006] In the case where the access to a transmission is to be restricted, for example, in a pay TV system, conditional access data may be included in a table or section broadcast in the transport stream with the transmission. This conditional access data is filtered by the receiver/decoder and passed to a portable security module, such as smart card, inserted in the decoder. The data is then processed by the smart card in order to generate, for example, a control word subsequently used by the decoder to descramble a transmission.

30 [0007] One problem with known systems lies in the volume of data that will be received and processed by the receiver/decoder and notably the volume of conditional access messages eventually forwarded to the smart card or security module. In particular, the processing capabilities of a smart card processor and the capacity of the communication channel between the decoder and smart card may be insufficient to handle a given volume of messages. This problem is exacerbated by the increasing tendency for programmes to be transmitted with multiple conditional access messages enabling access by different operators to the same programme (e.g. a football match or a thematic television channel).

35 [0008] According to the present invention, there is provided a decoder for a digital transmission system adapted to receive a transport packet stream containing table, section or other packetised data encapsulated within the packet payloads and characterised in that the decoder comprises a means for filtering the encapsulated data configurable in response to filter data received from a portable security module.

40 [0009] Filtering data at the table or section level in response to information from the security module enables a more precise identification and selection of data to be carried out, for example, to extract relevant conditional access messages addressed to the module. In practice, and as will be described below, this filtering at the table or section level may be carried out after and in addition to a filtering carried out at the transport packet level.

45 [0010] Preferably, the means for filtering encapsulated data is configurable in response to filter data comprising at least a table ID or section ID value transmitted by the portable security module. The means for filtering encapsulated data may equally be configurable in accordance with other data received from the portable security module.

[0011] In a preferred embodiment, the means for filtering encapsulated data is further adapted to forward to the security module conditional access data obtained in accordance with the filter data received from the security module.

50 [0012] Whilst the present invention is particularly adapted to enable a reduction of the volume of conditional access messages communicated between the decoder and the module, it will be nevertheless appreciated that the encapsulated data may be configured by the security module to extract data other than conditional access data and having a destination other than the security module.

[0013] Conditional access data filtered and forwarded to the security module may comprise entitlement control messages (ECMs) and/or entitlement management messages (EMMs).

55 [0014] Even within a group of messages associated with a single conditional access system there may be a large number of messages irrelevant to a particular user within that system. For example, within a single conditional access system a number of different groups of users may be defined leading to the generation of a number of EMMs, not all of which may be relevant to a given user.

[0015] Preferably therefore, filter data provided by the security module comprises data used by the filter means to

extract group and/or individual entitlement management messages addressed to the security module.

[0016] In one embodiment, the decoder is adapted to receive a control word generated by the security module in response to the conditional access data forwarded thereto, the control word being used by the decoder to descramble a scrambled transmission.

5 [0017] In addition to a filtering at the table or section level, the decoder may further carry out a transport level filtering in order, for example, to extract only these packets comprising data associated with the particular conditional access system used by the security module. Preferably, therefore the decoder further comprises a means for filtering transport packet data configurable in response to data received from the security module.

[0018] Advantageously, the means for filtering transport packet data may be configurable in response to data representing the identity of the conditional access system received from the security module.

10 [0019] In one embodiment, the transport packet filtering means is adapted to extract transport packets containing a program map table and a conditional access table, the decoder further comprising selection means adapted to receive the program map table and conditional access table from the transport packet filtering means and conditional access identity data from the security module and thereafter configure the transport packet filtering means to extract transport packet data associated with the conditional access system in question.

15 [0020] In order to preserve security in the system, some or all communications between the security module and the decoder may be encrypted. In particular, the descrambling control word generated by the security module and eventually transmitted to the decoder may be encrypted.

[0021] The present invention has been described above in relation to a decoder. Other aspects of the invention relate to a method of filtering encapsulated data in a transport packet stream and a security module for use with a decoder or method of the present invention. In one embodiment, the security module may conveniently comprise a smart card.

[0022] Whilst the present invention may apply to any packet transmission system comprising a transport stream layer and a table or section layer, the present invention is particularly applicable to a decoder adapted to receive an MPEG compatible data stream.

25 [0023] In this regard, the term "table, section or other packetised data" refers in its broadest sense to any data table, alone or in a sequence, and comprising a header and payload and that is itself encapsulated within a transport packet stream. As will be described in the preferred embodiment, the present invention is particularly applicable to filtering of data contained within an MPEG table, notably a single MPEG short form table. Other embodiments are nevertheless conceivable, for example, in which filtering is carried out on PES packets encapsulated within the transport packet payloads.

30 [0024] In the context of this application, the term MPEG refers to the data transmission standards developed by the International Standards Organisation working group "Motion Pictures Expert Group" and in particular but not exclusively the MPEG-2 standard developed for digital television applications and set out in the documents ISO 13818-1, ISO 13818-2, ISO 13818-3 and ISO 13818-4. In the context of the present patent application, the term MPEG includes all variants, modifications or developments of MPEG formats applicable to the field of digital data transmission.

35 [0025] As used herein, the term "smart card" includes, but not exclusively so, any chip-based card device, or object of similar function and performance, possessing, for example, microprocessor and/or memory storage. Included in this term are devices having alternative physical forms to a card, for example key-shaped devices such as are often used in TV decoder systems.

40 [0026] The term "decoder" or "receiver/decoder" used herein may connote a receiver for receiving either encoded or non-encoded signals, for example, television and/or radio signals, which may be broadcast or transmitted by some other means. Embodiments of such receiver/decoders may include a decoder integral with the receiver for decoding the received signals, for example, in a "set-top box", a decoder functioning in combination with a physically separate receiver, as well as a decoder including additional functions, such as a web browser or integrated with a video recorder or a television.

45 [0027] As used herein, the term "digital transmission system" includes any transmission system for transmitting or broadcasting digital data, for example primarily audiovisual or multimedia digital data. Whilst the present invention is particularly applicable to a broadcast digital television system, the invention may also be applicable to a fixed telecommunications network for multimedia internet applications, to a closed circuit television, and so on.

50 [0028] As used herein, the term "digital television system" includes for example any satellite, terrestrial, cable and other system.

[0029] There will now be described, by way of example only, a preferred embodiment of the invention, with reference to the following figures, in which:

55 Figure 1 shows the overall architecture of a digital TV system according to this embodiment;

Figure 2 shows the architecture of the conditional access system of Figure 1;

Figure 3 shows the hierarchy of MPEG-2 packets, in particular those associated with conditional access messages;

Figure 4 shows the structure of long form and short form MPEG-2 private sections;

Figure 5 shows the elements of a receiver/decoder for use in this embodiment;

Figure 6 shows the elements of the receiver/decoder used to process the transport stream, in particular in relation to conditional access messages; and

Figure 7 shows the structure of the PID and section filters of the filter unit of Fig. 6.

[0030] An overview of a digital television broadcast and reception system 1 is shown in Figure 1. The invention includes a mostly conventional digital television system 2 which uses the MPEG-2 compression system to transmit compressed digital signals. In more detail, MPEG-2 compressor 3 in a broadcast centre receives a digital signal stream (for example a stream of audio or video signals). The compressor 3 is connected to a multiplexer and scrambler 4 by linkage 5. The multiplexer 4 receives a plurality of further input signals, assembles one or more transport streams and transmits compressed digital signals to a transmitter 6 of the broadcast centre via linkage 7, which can of course take a wide variety of forms including telecom links.

[0031] The transmitter 6 transmits electromagnetic signals via uplink 8 towards a satellite transponder 9, where they are electronically processed and broadcast via a national downlink 10 to earth receiver 11, conventionally in the form of a dish owned or rented by the end user. The signals received by receiver 11 are transmitted to an integrated receiver/decoder 12 owned or rented by the end user and connected to the end user's television set 13. The receiver/decoder 12 decodes the compressed MPEG-2 signal into a television signal for the television set 13.

[0032] A conditional access system 20 is connected to the multiplexer 4 and the receiver/decoder 12, and is located partly in the broadcast centre and partly in the decoder. It enables the end user to access digital television broadcasts from one or more broadcast suppliers. A smartcard, capable of decrypting messages relating to commercial offers (that is, one or several television programmes sold by the broadcast supplier), can be inserted into the receiver/decoder 12. Using the decoder 12 and smartcard, the end user may purchase events in either a subscription mode or a pay-per-view mode.

[0033] An interactive system 17, also connected to the multiplexer 4 and the receiver/decoder 12 and again located partly in the broadcast centre and partly in the decoder, may be provided to enable the end user to interact with various applications via a modemed back channel 16.

[0034] The conditional access system 20 will now be described in more detail.

[0035] With reference to Figure 2, in overview the conditional access system 20 includes a Subscriber Authorization System (SAS) 21. The SAS 21 is connected to one or more Subscriber Management Systems (SMS) 22, one SMS for each broadcast supplier, by a respective TCP-IP linkage 23 (although other types of linkage could alternatively be used). Alternatively, one SMS could be shared between two broadcast suppliers, or one supplier could use two SMSs, and so on.

[0036] First encrypting units in the form of ciphering units 24 utilising "mother" smartcards 25 are connected to the SAS by linkage 26. Second encrypting units again in the form of ciphering units 27 utilising mother smartcards 28 are connected to the multiplexer 4 by linkage 29. The receiver/decoder 12 receives a "daughter" smartcard 30. It is connected directly to the SAS 21 by Communications Servers 31 via the modemed back channel 16. The SAS sends, amongst other things, subscription rights to the daughter smartcard on request.

[0037] The smartcards contain the secrets of one or more commercial operators. The "mother" smartcard encrypts different kinds of messages and the "daughter" smartcards decrypt the messages, if they have the rights to do so.

[0038] The first and second ciphering units 24 and 27 comprise a rack, an electronic VME card with software stored on an EEPROM, up to 20 electronic cards and one smartcard 25 and 28 respectively, for each electronic card, one card 28 for encrypting the ECMs and one card 25 for encrypting the EMMs.

[0039] The operation of the conditional access system 20 of the digital television system will now be described in more detail with reference to the various components of the television system 2 and the conditional access system 20.

Multiplexer and Scrambler

[0040] With reference to Figures 1 and 2, in the broadcast centre, the digital audio or video signal is first compressed (or bit rate reduced), using the MPEG-2 compressor 3. This compressed signal is then transmitted to the multiplexer and scrambler 4 via the linkage 5 in order to be multiplexed with other data, such as other compressed data.

[0041] The scrambler generates a control word used in the scrambling process and included in the MPEG-2 stream in the multiplexer. The control word is generated internally and enables the end user's integrated receiver/decoder 12

to descramble the programme.

[0042] Access criteria, indicating how the programme is commercialised, are also added to the MPEG-2 stream. The programme may be commercialised in either one of a number of "subscription" modes and/or one of a number of "Pay Per View" (PPV) modes or events. In the subscription mode, the end user subscribes to one or more commercial offers, or "bouquets", thus getting the rights to watch every channel inside those bouquets. In the preferred embodiment, up to 960 commercial offers may be selected from a bouquet of channels.

[0043] In the Pay Per View mode, the end user is provided with the capability to purchase events as he wishes. This can be achieved by either pre-booking the event in advance ("pre-book mode"), or by purchasing the event as soon as it is broadcast ("impulse mode"). In the preferred embodiment, all users are subscribers, whether or not they watch in subscription or PPV mode, but of course PPV viewers need not necessarily be subscribers.

Entitlement Control Messages

[0044] Both the control word and the access criteria are used to build an Entitlement Control Message (ECM). This is a message sent in relation with a scrambled program; the message contains a control word (which allows for the descrambling of the program) and the access criteria of the broadcast program. The access criteria and control word are transmitted to the second encrypting unit 27 via the linkage 29. In this unit, an ECM is generated, encrypted and transmitted on to the multiplexer and scrambler 4. During a broadcast transmission, the control word typically changes every few seconds, and so ECMs are also periodically transmitted to enable the changing control word to be descrambled. For redundancy purposes, each ECM typically includes two control words; the present control word and the next control word.

[0045] Each service broadcast by a broadcast supplier in a data stream comprises a number of distinct components; for example a television programme includes a video component an audio component, a sub-title component and so on. Each of these components of a service is individually scrambled and encrypted for subsequent broadcast to the transponder 9. In respect of each scrambled component of the service, a separate ECM is required. Alternatively, a single ECM may be required for all of the scrambled components of a service. Multiple ECMs are also generated in the case where multiple conditional access systems control access to the same transmitted program.

Programme Transmission

[0046] The multiplexer 4 receives electrical signals comprising encrypted EMMs from the SAS 21, encrypted ECMs from the second encrypting unit 27 and compressed programmes from the compressor 3. The multiplexer 4 scrambles the programmes and sends the scrambled programmes, the encrypted EMMs and the encrypted ECMs to a transmitter 6 of the broadcast centre via the linkage 7. The transmitter 6 transmits electromagnetic signals towards the satellite transponder 9 via uplink 8.

Programme Reception

[0047] The satellite transponder 9 receives and processes the electromagnetic signals transmitted by the transmitter 6 and transmits the signals on to the earth receiver 11, conventionally in the form of a dish owned or rented by the end user, via downlink 10. The signals received by receiver 11 are transmitted to the integrated receiver/decoder 12 owned or rented by the end user and connected to the end user's television set 13. The receiver/decoder 12 demultiplexes the signals to obtain scrambled programmes with encrypted EMMs and encrypted ECMs.

[0048] If the programme is not scrambled, that is, no ECM has been transmitted with the MPEG-2 stream, the receiver/decoder 12 decompresses the data and transforms the signal into a video signal for transmission to television set 13.

[0049] If the programme is scrambled, the receiver/decoder 12 extracts the corresponding ECM from the MPEG-2 stream and passes the ECM to the "daughter" smartcard 30 of the end user. This slots into a housing in the receiver/decoder 12. The daughter smartcard 30 controls whether the end user has the right to decrypt the ECM and to access the programme. If not, a negative status is passed to the receiver/decoder 12 to indicate that the programme cannot be descrambled. If the end user does have the rights, the ECM is decrypted and the control word extracted. The decoder 12 can then descramble the programme using this control word. The MPEG-2 stream is decompressed and translated into a video signal for onward transmission to television set 13.

Entitlement Management Messages (EMMs)

[0050] The EMM is a message dedicated to an individual end user (subscriber), or a group of end users. Each group may contain a given number of end users. This organisation as a group aims at optimising the bandwidth; that is,

access to one group can permit the reaching of a great number of end users.

[0051] Various specific types of EMM can be used. Individual EMMs are dedicated to individual subscribers, and are typically used in the provision of Pay Per View services; these contain the group identifier and the position of the subscriber in that group.

5 [0052] Group subscription EMMs are dedicated to groups of, say, 256 individual users, and are typically used in the administration of some subscription services. This EMM has a group identifier and a subscribers' group bitmap.

[0053] Audience EMMs are dedicated to entire audiences, and might for example be used by a particular operator to provide certain free services. An "audience" is the totality of subscribers having smartcards which bear the same conditional access system identifier (CA ID). Finally, a "unique" EMM is addressed to the unique identifier of the smartcard.

10 Subscriber Management System (SMS)

[0054] A Subscriber Management System (SMS) 22 includes a database 32 which manages, amongst others, all of the end user files, commercial offers, subscriptions, PPV details, and data regarding end user consumption and authorization. The SMS may be physically remote from the SAS.

15 [0055] Each SMS 22 transmits messages to the SAS 21 via respective linkage 23 which imply modifications to or creations of Entitlement Management Messages (EMMs) to be transmitted to end users.

[0056] The SMS 22 also transmits messages to the SAS 21 which imply no modifications or creations of EMMs but imply only a change in an end users state (relating to the authorization granted to the end user when ordering products or to the amount that the end user will be charged).

20 [0057] The SAS 21 sends messages (typically requesting information such as call-back information or billing information) to the SMS 22, so that it will be apparent that communication between the two is two-way.

25 Subscriber Authorization System (SAS)

[0058] The messages generated by the SMS 22 are passed via linkage 23 to the Subscriber Authorization System (SAS) 21, which in turn generates messages acknowledging receipt of the messages generated by the SMS 21 and passes these acknowledgements to the SMS 22.

30 [0059] In overview the SAS comprises a Subscription Chain area to give rights for subscription mode and to renew the rights automatically each month, a Pay Per View Chain area to give rights for PPV events, and an EMM injector for passing EMMs created by the Subscription and PPV chain areas to the multiplexer and scrambler 4, and hence to feed the MPEG stream with EMMs. If other rights are to be granted, such as Pay Per File (PPF) rights in the case of downloading computer software to a user's Personal Computer, other similar areas are also provided.

35 [0060] One function of the SAS 21 is to manage the access rights to television programmes, available as commercial offers in subscription mode or sold as PPV events according to different modes of commercialisation (pre-book mode, impulse mode). The SAS 21, according to those rights and to information received from the SMS 22, generates EMMs for the subscriber.

40 [0061] The EMMs are passed to the Cipharing Unit (CU) 24 for cipharing with respect to the management and exploitation keys. The CU completes the signature on the EMM and passes the EMM back to a Message Generator (MG) in the SAS 21, where a header is added. The EMMs are passed to a Message Emitter (ME) as complete EMMs. The Message Generator determines the broadcast start and stop time and the rate of emission of the EMMs, and passes these as appropriate directions along with the EMMs to the Message Emitter. The MG only generates a given EMM once; it is the ME which performs cyclic transmission of the EMMs.

45 [0062] On generation of an EMM, the MG assigns a unique identifier to the EMM. When the MG passes the EMM to the ME, it also passes the EMM ID. This enables identification of a particular EMM at both the MG and the ME.

[0063] In systems such as simulcrypt which are adapted to handle multiple conditional access systems e.g. associated with multiple operators, EMM streams associated with each conditional access system are generated separately and multiplexed together by the multiplexer 4 prior to transmission.

50 Conditional Access Messages in the Transport Stream

[0064] The different nature of ECM and EMM messages leads to differences vis à vis the mode of transmission of the messages in the MPEG transport stream. ECM messages, which carry the control words needed to descramble a programme are necessarily linked to the video and audio streams of the programme being transmitted, in contrast EMM messages are general messages broadcast asynchronously to transmit rights information to individual or groups of customers. This difference is reflected in the placing of ECM and EMM messages within the MPEG transport stream.

55 [0065] As is known, MPEG transport packets are of a fixed length of 188 bytes including a header. In a standard packet, the three bytes of the header following the synchronisation data comprise:

TABLE I

Transport error indicator	1 bit
Payload unit indicator	1 bit
Transport priority	1 bit
PID	13 bits
Transport scrambling control	2 bits
Adaptation field control	2 bits
Continuity counter	4 bits

[0066] The characteristics of these fields are largely determined by the MPEG standard.

[0067] Referring to Figure 3, the organisation of data within a transport stream will be described. As shown, the transport stream contains a programme association table 40 ("PAT"), the PID in the header of the packet being fixed by the MPEG-2 standard at a value of 0x00. The programme access table 40 provides the entry point for access to programme data and contains a table referring to the PID values of the programme map tables ("PMT") 41, 42 associated with a number of programmes. Each programme map table 41, 42 contains in turn a reference to the PID values of the packet streams of the audio tables 43 and video tables 44 of that programme.

[0068] As shown, the programme map table 42 also contains references to the PID values of other packets 45, 46 containing additional data relating to the programme in question. In the present case ECM data generated by a number of conditional access systems and associated with the programme in question is contained within the referred packets 45, 46.

[0069] In addition to the programme access table PAT 40, the MPEG transport stream further comprises a conditional access table 47 ("CAT"), the PID value of which is fixed at 0x01. Any packet headers containing this PID value are thus automatically identified as containing access control information. The CAT table 47 refers to the PID values of MPEG packets 48, 49, 50 associated with EMM data associated with one or more conditional access systems. As with the PMT packets, the PID values of the EMM packets referred to in the CAT table are not fixed and may be determined at the choice of the system operator.

Private Section Data

[0070] In conformity with the MPEG-2 standard, information contained with a packet payload is subject to a further level of structure according to the type of data being transported. In the case of audio, visual, teletext, subtitle or other such rapidly evolving and synchronised data, the information is assembled in the form of what is known as a packetised elementary stream or PES. This data stream, which is formed by assembling the payloads of the transmitted packets, itself comprises a sequence of packets, each packet comprising a packet header and payload. Unlike the transmitted packets in the transport stream, the length of PES packets is variable.

[0071] In the case of other data, such as application data or, in this example, ECM and EMM data, a different format from PES packeting is proscribed. In particular, data contained in the transport packet payload is divided into a series of sections or tables, the table or section header including a table ID or TID identifying the table in question. Depending on the size of the data, a section may be contained entirely within a packet payload or may be extended in a series of tables over a number of transport packets. In the MPEG-2 context, the term "table" is often used to refer to a single table of data, whilst "section" refers to one of a plurality of tables with the same TID value.

[0072] As with transport packet data and PES packet data, the data structure of a table or section is additionally defined by the MPEG-2 standard. In particular, two possible syntax forms for private table or section data are proposed; a long form or a short form, as illustrated in Figure 4.

[0073] In both the short and long form, the header includes at least the data 60 comprising:

TABLE II

Table id	8 bits
Section syntax indicator	1 bit

TABLE II (continued)

Private indicator/reserved	1 bit
ISO reserved	2 bits
Section length	12 bits

[0074] The private indicator and private section lengths are comprised of data not fixed by the MPEG-2 standard and which may be used by the system operator for his own purposes.

[0075] In the case of short form, the header 60 is immediately followed by the payload data 61. In the case of the long form, a further header section 62 is provided before the payload 63 and the message equally includes a CRC check value 64. The long form, which is typically used when a message is so long that it must be divided into a number of sections, contains the information necessary to assemble the sections, such as the section number, the number of the last section in the sequence of sections etc.

[0076] For further information regarding the long and short form table data, the reader is directed to the MPEG-2 standard.

[0077] In the case of conditional access ECM and EMM messages, the data may usually be accommodated in a single table and the short form will be the appropriate format. A specific syntax for such short form conditional access messages is proposed in the context of the present invention, namely:

TABLE III

Table id (filter data)	8 bits (1 byte)
Section syntax indicator	1 bit
Private indicator/reserved	1 bit
ISO reserved	2 bits
Section length	12 bits
CA specific header field (filter data)	56 bits (7 bytes)

[0078] For such CA messages, the table id value may be set by the system operator at, for example, 0x80 and 0x81 for ECM messages (for example, odd and even messages) and 0x82 to 0x8F for EMM messages. These values are not MPEG-2 proscribed and may be chosen at the discretion of the system operator.

[0079] Equally, in the case of the CA specific header field, hereby designated as the first 7 bytes of the payload following the header, the parameters may be set by the system operator to reflect, for example, the fact that the CA message is an EMM message carrying individual, group or audience subscription information. In this manner the "header" of such a table or section is extended.

[0080] The advantages of such message syntax will become clear later, with regard to the processing and filtering of messages by the receiver/decoder, notably by using the Table id and CA specific field data.

Receiver/decoder

[0081] Referring to Figure 5, the elements of a receiver/decoder 12 or set-top box for use in a digital broadcast system and adapted to be used in the present invention will now be described. As will be understood, the basic elements of this decoder are largely conventional and their implementation will be within the capabilities of one skilled in the art.

[0082] As shown, the decoder 12 is equipped with several interfaces for receiving and transmitting data, in particular a tuner 70 for receiving broadcast MPEG transmissions, a serial interface 71, a parallel interface 72, and a modem 73 for sending and receiving data via the telephone network. The decoder also includes a first and second smart card reader 74 and 75, the first reader 74 for accepting the subscription smart card and the second reader 75 for accepting bank and/or other smart cards.

[0083] The decoder also includes a receiver 76 for receiving infra-red control signals from a handset remote control 77 and a Peritel output for sending audiovisual signals to a television 13 connected to the decoder.

[0084] Processing of digital signals received via the interfaces and generation of output signals is handled by an ensemble of hardware and software elements here grouped together as a central control unit 78. The software architecture of the control unit within the decoder may correspond to that used in a known decoder and will not be described here in any detail. It may be based, for example, on a virtual machine interacting via an interface layer with a lower level

operating system implemented in the hardware components of the decoder. In terms of hardware architecture, the control unit 78 will be equipped with a processor, memory elements such as ROM, RAM, FLASH memory etc. as in known decoders.

[0085] Applications processed by the control unit 78 may be resident applications stored in the ROM or FLASH of the decoder or applications broadcast and downloaded via the MPEG interface 2 of the decoder. Applications can include program guide applications, games, interactive services, teleshopping applications, as well as initiating applications to enable the decoder to be immediately operational upon start-up and applications for configuring aspects of the decoder. Applications are stored in memory locations in the decoder and represented as resource files comprising graphic object descriptions files, unit files, variables block files, instruction sequence files, applications files, data files etc.

Filtering of Conditional Access Data

[0086] Figure 6 shows in schematic form the elements necessary for processing packet and table data in accordance with this embodiment of the invention. As will be understood, the elements shown in this figure may be implemented in hardware, software or in combination of the two.

[0087] The broadcast transmission received from the satellite receiver are passed via the conventional tuner 70 and an associated demodulator unit 79. The tuner 70 typically scans a range of frequencies, stopping when a chosen carrier frequency is detected within that range. The signals are then treated by the demodulator unit 79 which extracts and forwards the transport packet stream to a demux and filter unit 80. The filter structure of the demux and filter unit 80 will be described in detail below in relation to Figure 7. As will be understood, the actual choice of components needed to implement such a unit is at the discretion of the manufacturer and the most important aspect of such a unit is the chosen filter configuration.

[0088] In the case of data encrypted in accordance with a conditional access system as per the present embodiment, the filter unit interacts with a smart card 30 (or any other secure device) inserted in the decoder 12 and a channel parameter application 81, typically implemented as a software application in the decoder.

[0089] The filter unit 80 extracts from the transport packet stream the PMT and CAT tables present in the stream. Referring back to Figure 3, this filtering operation is carried out at a PID level, the CAT table being identified by the PID value 0x01 and the appropriate PMT table corresponding to the chosen broadcast channel being extracted via the PAT table (PID value: 0x00) and the PID value of the chosen channel identified in the PAT table.

[0090] The channel parameter application 81 additionally receives from the smart card 30 an identification of the conditional access system associated with that smart card. Again, referring back to Figure 3, a first conditional access system is associated with ECM and EMM data in the packets 45 and 48, respectively. Using the conditional access system ID received from the smart card 30 and the PMT and CAT tables received from the filter unit 80, the application 81 determines the PID values of the conditional access packets associated with the conditional access system in question and returns these values to the filter unit 80.

[0091] In the case of a simplified system, where a relatively small number of ECM and EMMs are emitted, no other filtering may be necessary and these PID values may be used by the filter unit 80 to extract all relevant ECM and EMM private sections from the identified packets and to thereafter forward the data contained within these sections to the smart card 30.

[0092] This conditional access data is then processed by the microprocessor within the smart card 30 and the control word associated with the transmission passed to a descrambling unit 83. The descrambling unit 83 receives scrambled audiovisual or other data information extracted from the transport packet stream by the demux and filter unit 80, descrambles the information using the control word and thereafter passes the data to a convention MPEG-2 chip which prepares the data for subsequent display on the associated television display.

[0093] However, whilst a PID level filter enables an extraction of those ECM and EMM messages associated exclusively with the conditional access system in question, there may nevertheless be a large proportion of messages irrelevant to the user. These messages may include group EMM messages for other user groups, individual EMM messages for other users etc. The throughput of conditional access messages passed to the smart card may therefore be very high. Given the limitations of the processor power and memory of smart cards, this throughput may be in practice more than the card can handle.

[0094] In order to overcome this problem, the smartcard 30 is adapted to pass further filter data to the unit 80 for use in a section or table level filter process.

[0095] Referring to the Table III above, tables containing conditional access data include Table id and CA specific header fields which are chosen to identify, for example, the presence of an EMM or ECM (table id values 0x80 or 0x81 and 0x82 to 0x8F, respectively) and the type of message (CA specific data identifying the group concerned by a group EMM message, the presence of an audience EMM message etc.). Depending on the data that it requires, the smart card 30 will send the necessary table id and CA specific data to configure the filter unit to extract and return only those conditional access messages of interest to the smart card. In this way, the flow of data sent to the smart card may be

reduced to conform with the processing capabilities of the smart card microprocessor.

[0096] Referring to Figure 7, the details of the filtering unit 80 will be described. Typically, the unit may be implemented as a hardware resource, driven by a firmware managing application with the receiver/decoder. As shown, a first set of filters 85 carries out a PID filtering process using the CA PID information received from the channel parameter application. The PID filters 85 may equally be configured to extract other relevant packets such as the PMT, CAT tables sent to the channel parameter application. Other PID filters (not shown) may be used to extract the audiovisual PES packet information eventually sent to the descrambler etc.

[0097] Once stripped of the packet header, the private section or table data is then routed to a set of prefilters 86 adapted to filter the 8 bytes in the extended header of a table. As shown in Table III, 1 byte of the extended header is associated with the table id, 7 bytes with the CA specific information. The filtering operation is carried out by comparison of the 8 byte pattern in a table with the filter data received from the smart card. Some bits within the 8 byte, 64 bit pattern may be masked or ignored in the evaluation. In this embodiment, 32 different patterns are proposed, a subset of these patterns being applied by the prefilters in dependence of the information received from the smart card. If one pattern matches, the section is sent to the FIFO buffer element 87. If no pattern matches, the section is ignored. The filters 86 equally act to extract from the appropriate sections the PMT and CAT table information, which is passed to a FIFO buffer 88.

[0098] Due to the characteristics of the transport layer, the arrival of sections is bursty. The buffer capacity of the buffers 87, 88 must be sufficient to handle an average rate of 5Mbits/s, with the insertion of packets being based on a regular allocation with a possible deviation of $\pm 25\%$.

[0099] In order to better understand the invention, a proposed example of operating instructions handled by the section filters 86 will now be outlined.

Filter_all_sections (Filter_id, Target, Mask, Trigger_conditions, p/n)

This command retrieves every section matching the target except masked bits after trigger_conditions occurred.

Filter_next_section (Filter_id, Target, Mask, Trigger_conditions, p/n)

This command retrieves the next section matching the target except masked bits after trigger_conditions occurred. Trigger_conditions are related to other filters previously identified as matching.

Filter_id is an index between 0 and 31, pointing to a filter and an output queue. In addition, it gives the queueing priority, 0 being the highest priority.

Target is an 8 bytes pattern.

Mask is an 8 bytes pattern showing the bits to be masked in the target, value 0 means masked.

Trigger_conditions is a 32 bit bitmap, ORing filter_id triggering that filter. Bit set at 0 means no trigger condition. Self trigger condition is ignored.

p/n is a value, normally set to 1, positive for normal operation as described above. When set to 0 it means negative filtering, i.e., retrieve sections not matching target.

Examples of use:

Example 1:

[0100]

Filter_all_sections(5, 0x8C7C453AA8BFF00, 0xFF557FFFEFFFFFF00, 0, 1) will capture all EMMs corresponding To matching criteria.

Example 2:

[0101]

Filter_next_section(0, 0x8000000000000000, 0xFF00000000000000, 0, 1)

Filter_next_section(1, 0x8100000000000000, 0xFF00000000000000, 5, 1)

Filter_next_section(2, 0x8000000000000000, 0xFF00000000000000, 3, 1)

will start an ECM capture process with odd/even toggle.

Example 3:

[0102]

```

5   Filter_next_section(8, 0xPMT_TID0000Version_number00000000, 0xFF00001F00000000, 0, 0)
      Filter_next_section(1, 0x8100000000000000, 0xFF00000000000000, 0x14, 1)
      Filter_next_section(2, 0x8000000000000000, 0xFF00000000000000, 0x12, 1)
  
```

will start an ECM capture process with odd/even toggle, starting when there is a change in the PMT.

10 [0103] In terms of communication of CA messages and filter data to and from the smart card 82 and filter unit 80, a standard protocol such as ISO7816 may be used. Since not all of the data in the filtered private section is required by the smart card 82, the section may be modified and a message of the following format sent to the smart card:

15

Table id	8 bits
Zero	11 bits
Filter id	5 bits
CA specific header field	56 bits
CA message	N*8 bits

20

25 [0104] The meaning of each of these terms will be clear from the above description. In terms of the filter data sent from the smart card 82 to the filter 80, the following format may be used:

30

Number of filters	8 bits
Filtering instruction	5 bits
Filter id	5 bits
Target	64 bits
Mask	64 bits
Trigger conditions	5 bits
p/n	1 bit

35

40

Number_of_filters describes the number of filters to be set in this instruction.

45

Filtering_instruction is describing the type of instruction (filter next section, filter all sections).

Filter_id is an index pointing to a filter and an output queue. In addition, it gives the queueing priority, 0 being the highest priority.

Target is the target pattern.

Mask is a pattern showing the bits to be masked in the target, value 0 means masked.

50

Trigger_conditions is a bitmap. ORing filter_id triggering that filter. Bit set at 0 means no trigger condition. Self trigger condition is ignored.

p/n is a value, normally set to 1, positive for normal operation as described above. When set to 0 it means negative filtering, i.e., retrieve sections not matching target.

55

[0105] In practice, communications between the smart card and the receiver/decoder may be subject to a level of encryption or scrambling for security reasons. In particular, communications between the smart card 82 and filter unit 80, as well as the control word stream sent to the descrambler unit 83 may be encoded in this way. Encryption algorithms suitable for this purpose are widely known (RSA, DES etc.).

Claims

- 5 1. A decoder adapted to receive a transport packet stream containing table, section or other packetised data encapsulated within the packet payloads and characterised in that the decoder comprises a means for filtering the encapsulated data configurable in response to filter data received from a portable security module.
2. A decoder as claimed in claim 1 in which the means for filtering encapsulated data is configurable in response to filter data comprising at least a table ID or section ID value transmitted by the portable security module.
- 10 3. A decoder as claimed in claim 1 or 2 in which the means for filtering encapsulated data is further adapted to forward to the security module conditional access data obtained in accordance with the filter data received from the security module.
- 15 4. A decoder as claimed in claim 3 in which conditional access data forwarded to the security module comprises entitlement control messages (ECMs) and/or entitlement management messages (EMMs).
- 20 5. A decoder as claimed in claim 3 or 4 in which filter data provided by the security module comprises data used by the filter means to extract group and/or individual entitlement management messages addressed to the security module.
6. A decoder as claimed in any of claims 3 to 5 in which the decoder is adapted to receive a control word generated by the security module in response to the conditional access data forwarded thereto, the control word being used by the decoder to descramble a scrambled transmission.
- 25 7. A decoder as claimed in any preceding claim further comprising a means for filtering transport packet data configurable in response to data received from the security module.
8. A decoder as claimed in claim 7, in which the means for filtering transport packet data is configurable in response to data representing the identity of the conditional access system received from the security module.
- 30 9. A decoder as claimed in claim 8 in which the transport packet filtering means is adapted to extract transport packets containing a program map table and a conditional access table, the decoder further comprising selection means adapted to receive the program map table and conditional access table from the transport packet filtering means and conditional access identity data from the security module and thereafter configure the transport packet filtering means to extract transport packet data associated with the conditional access system in question.
- 35 10. A decoder as claimed in any preceding claim adapted to process encrypt and/or decrypt communications to and from the portable security module.
- 40 11. A security module for use with a decoder as claimed in any preceding claim and characterised in comprising a memory means for storing filter data subsequently communicated to the decoder to configure the means for filtering encapsulated data.
- 45 12. A security module as claimed in claim 13 comprising a smart card.
13. A method of processing a transport packet stream containing table, section or other packetised data encapsulated within the packet payloads characterised by receiving the transport stream in a decoder and filtering the encapsulated data in response to filter data received from a portable security module.
- 50 14. A method of processing a transport packet stream as claimed in claim 13 further comprising generating encapsulated data including conditional access data and filtering at the decoder using the encapsulated data and in response to filter data supplied by the portable security module.

55

Fig.1.

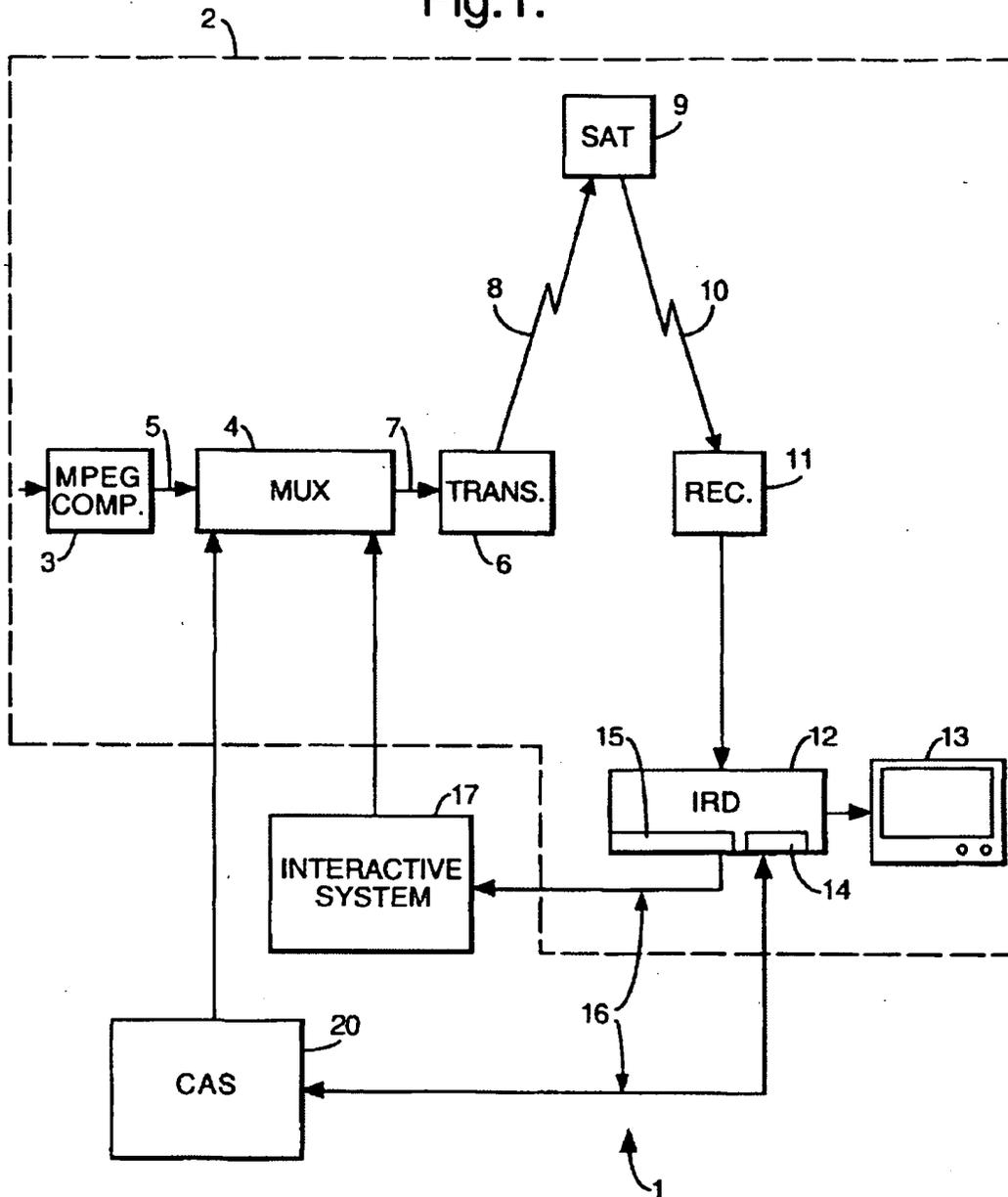


Fig.2.

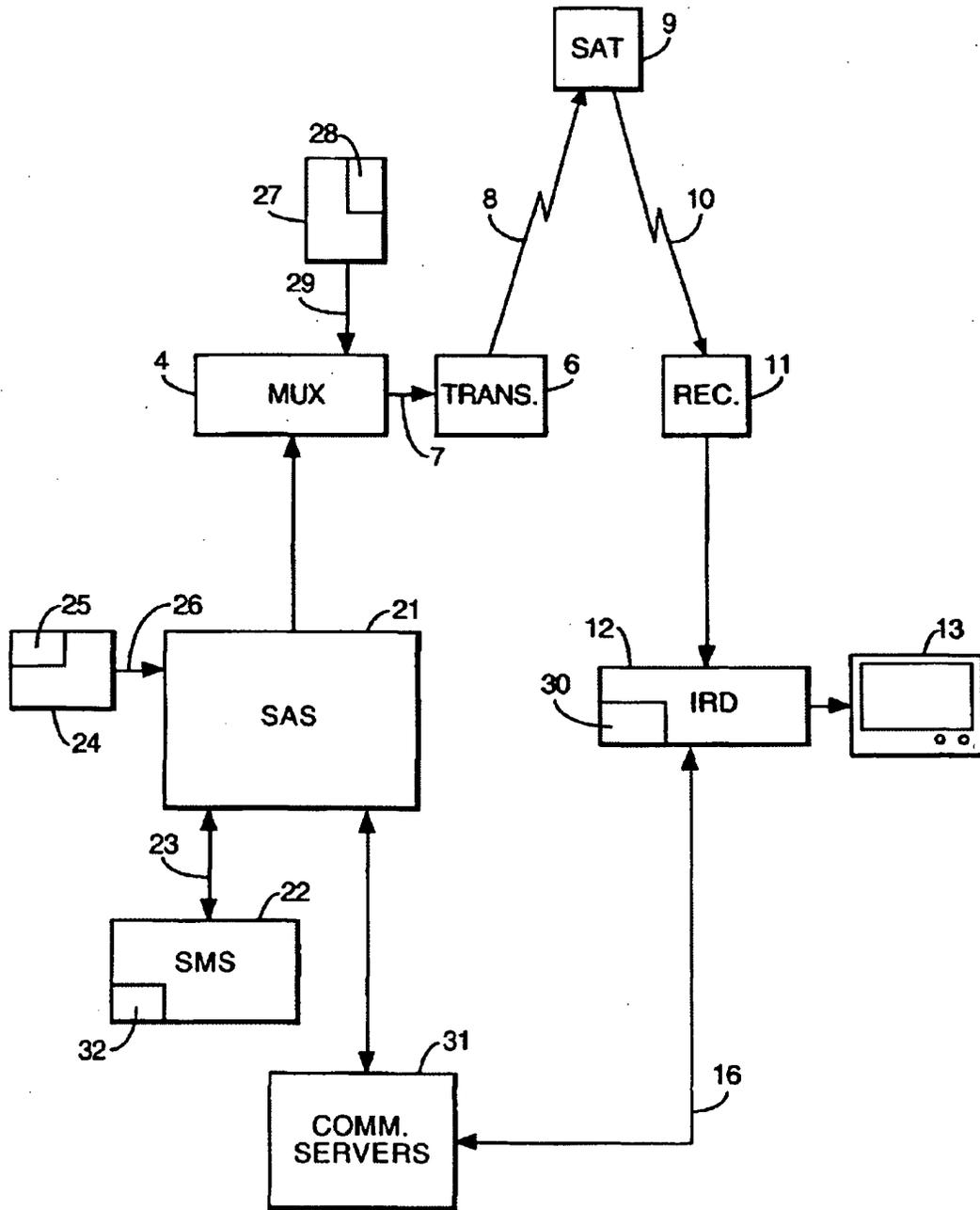


Fig.3.

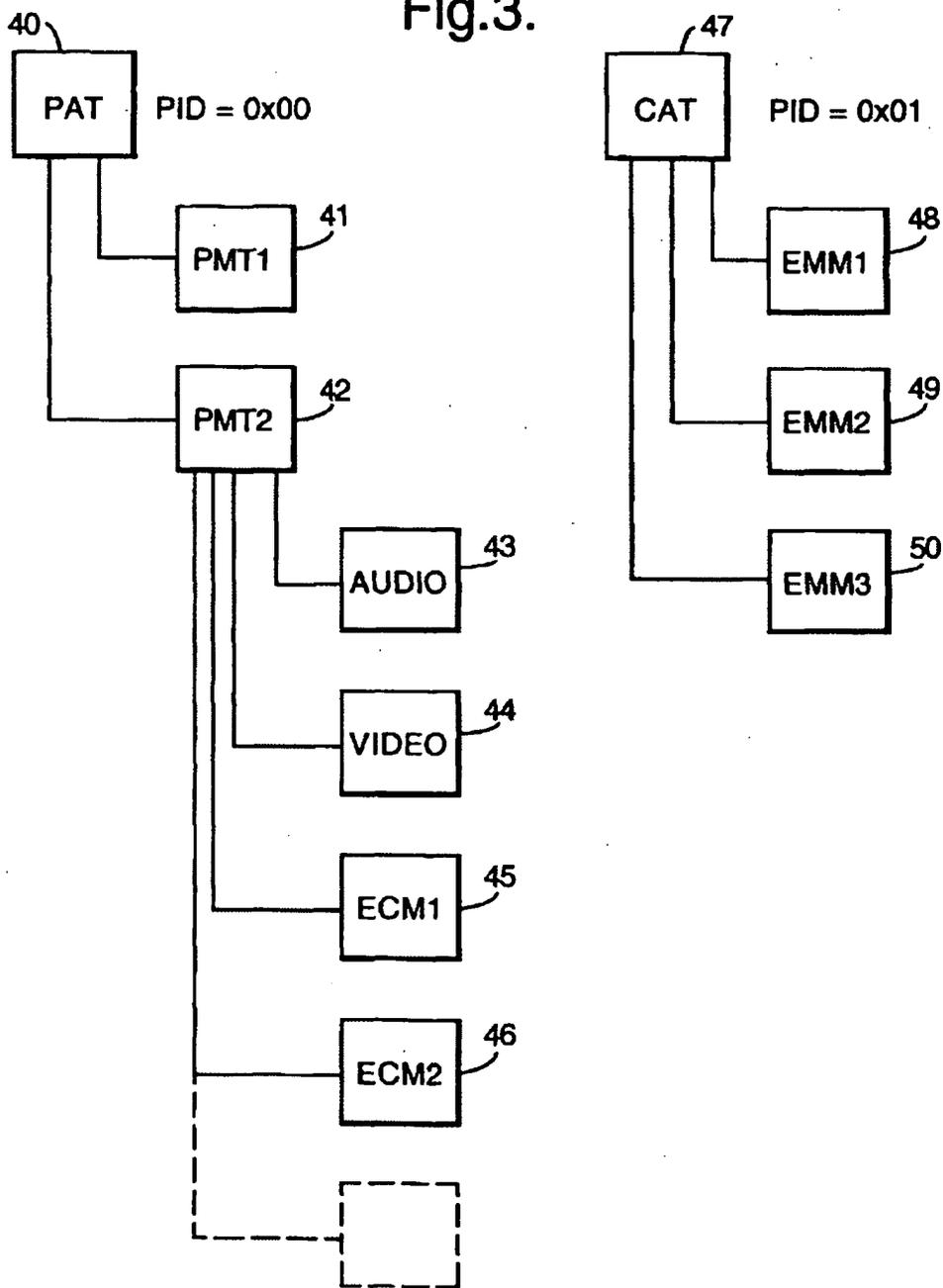


Fig. 4.

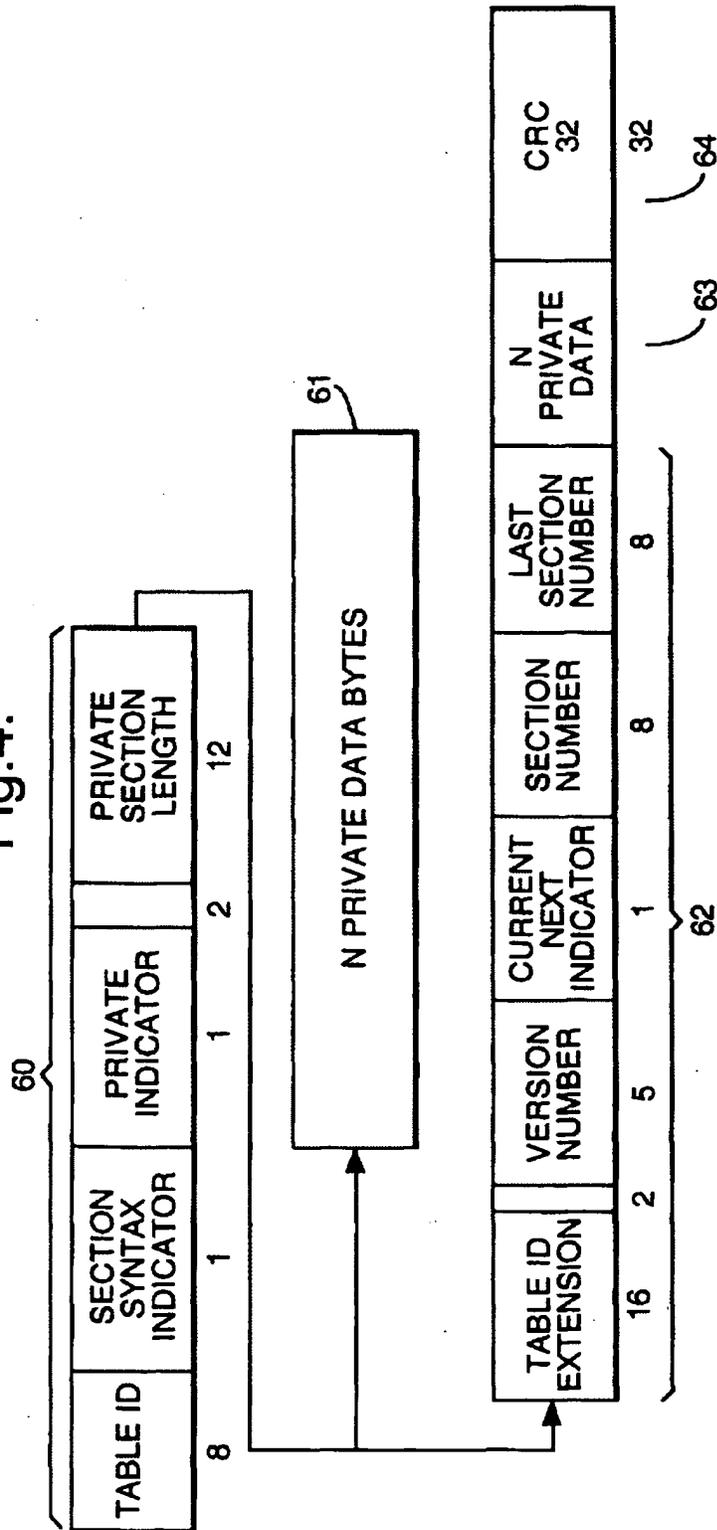


Fig.5.

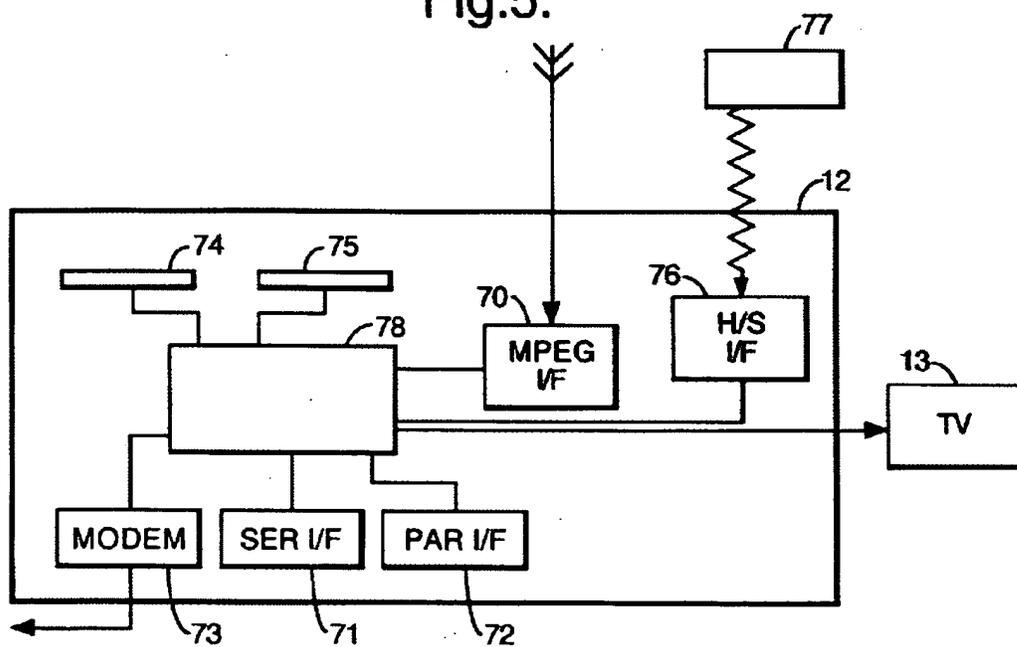
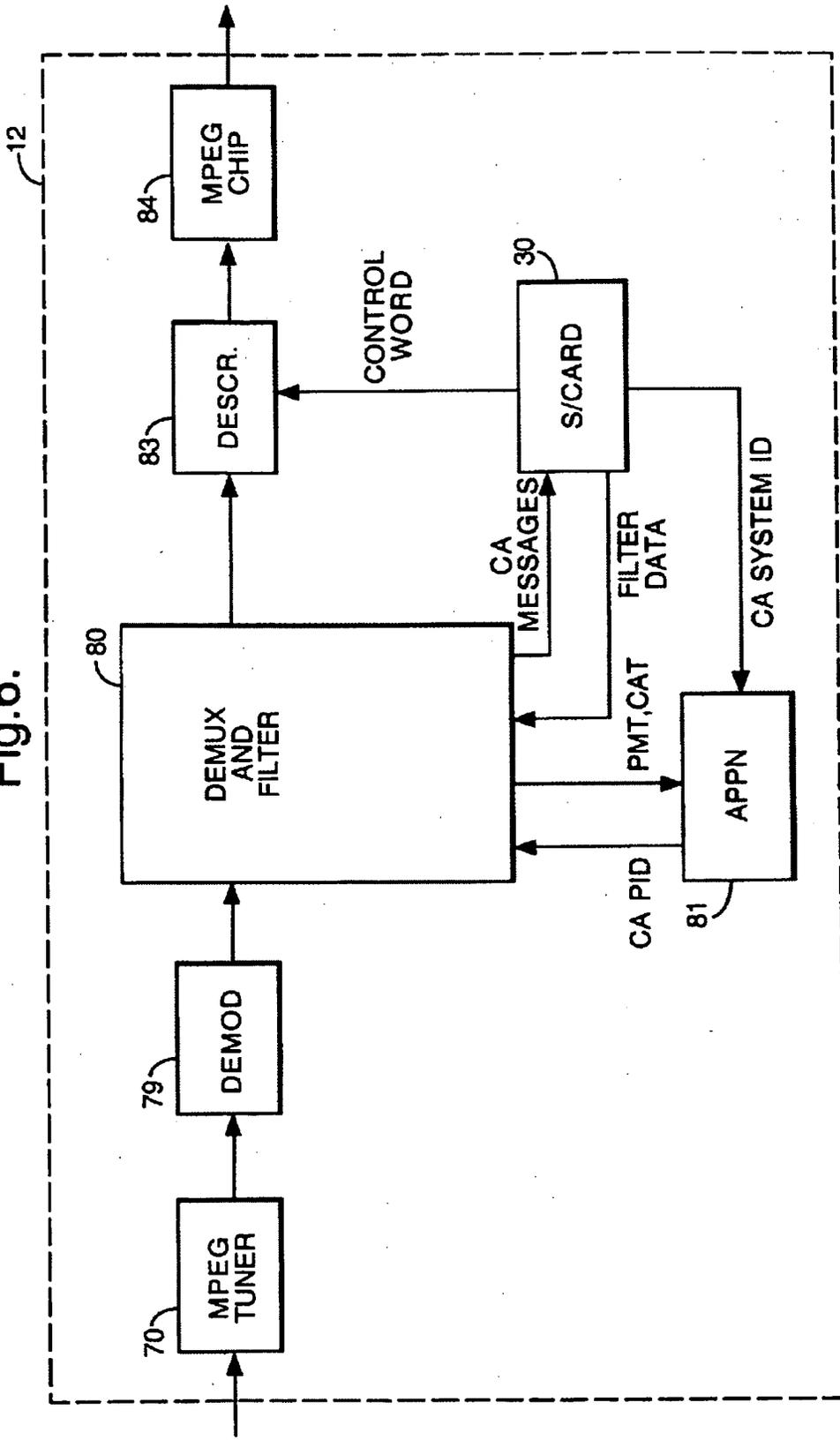
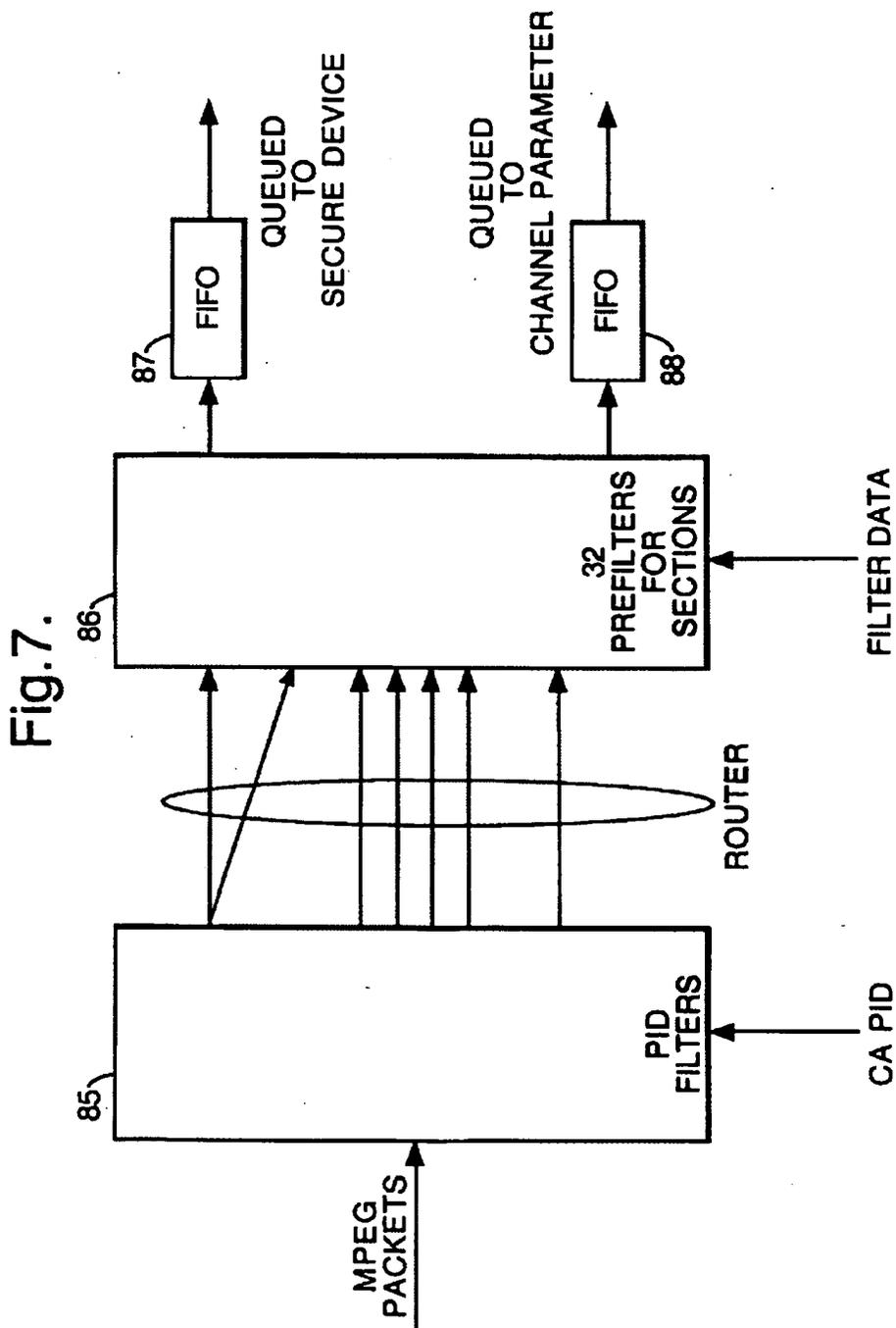


Fig.6.







European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 98 40 1374

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. CL.6)
X	WO 95 29560 A (THOMSON CONSUMER ELECTRONICS) 2 November 1995 * page 1, line 35 - page 2, line 25 * * page 4, line 23 - page 8, line 35 * * figure 3 *	1,3-5,8,10-14	H04N5/00
A	---	2,6,7,9	
X	WO 97 46008 A (THOMSON CONSUMER ELECTRONICS) 4 December 1997 * page 3, line 17 - page 10, line 9 *	1-3,6-14	
A	---	4,5	
A	"FUNCTIONAL MODEL OF A CONDITIONAL ACCESS SYSTEM" 21 December 1995, EBU REVIEW- TECHNICAL, NR. 266, PAGE(S) 64 - 77 XP000559450 * the whole document *	1-14	
A	SCHOONEVELD VAN D: "STANDARDIZATION OF CONDITIONAL ACCESS SYSTEMS FOR DIGITAL PAY TELEVISION" PHILIPS JOURNAL OF RESEARCH, vol. 50, no. 1/02, July 1996, pages 217-225, XP000627672 * page 218, line 12 - page 220, line 9 * -----	1-14	TECHNICAL FIELDS SEARCHED (Int. CL.6) H04N
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 3 November 1998	Examiner Fassnacht, C
CATEGORY OF CITED DOCUMENTS X: particularly relevant if taken alone Y: particularly relevant if combined with another document of the same category A: technological background O: non-written disclosure P: intermediate document		T: theory or principle underlying the invention E: earlier patent document, but published on, or after the filing date D: document cited in the application L: document cited for other reasons a: member of the same patent family, corresponding document	

EPO FORM 1503 01.92 (P04C01)

Booking by means of a virtual access ticket

Publication number: EP1103922
Publication date: 2001-05-30
Inventor: LAUTENSCHLAGER WOLFGANG (DE); STUERZ HEINZ (DE)
Applicant: CIT ALCATEL (FR)
Classification:
- international: **G06Q10/00; G07B15/00; G06Q10/00; G07B15/00;**
(IPC1-7): G07F7/08; G06F17/60; G07B15/00;
G07F17/42
- European:
Application number: EP20000124578 20001110
Priority number(s): DE19991056359 19991124

Also published as:

 EP1103922 (A3)
 DE19956359 (A1)

Cited documents:

 EP0950968
 US5598477
 NL9301902
 EP0713198
 GB2317258
more >>

Report a data error here

Abstract of EP1103922

The booking method has a reservation request received from a customer by a reservation agent, with the customer charge logged by the agent and an electrical signal containing coded data corresponding to an access authorisation transmitted back to the customer, for storage on an electronic data carrier, acting as a virtual entry ticket. Also included are Independent claims for the following: (a) a central server for a reservation booking method; (b) a computer program for a reservation booking method

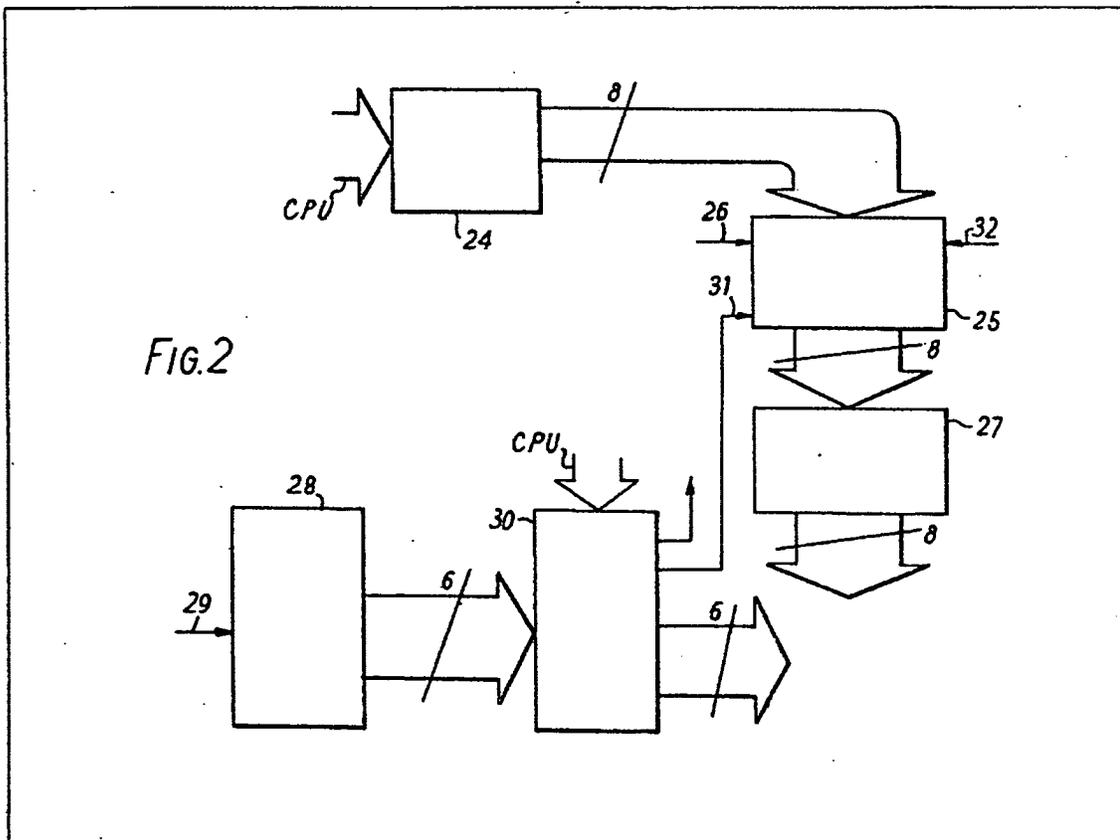
Data supplied from the *esp@cenet* database - Worldwide

(12) UK Patent Application (19) GB (11) 2 022 969 A

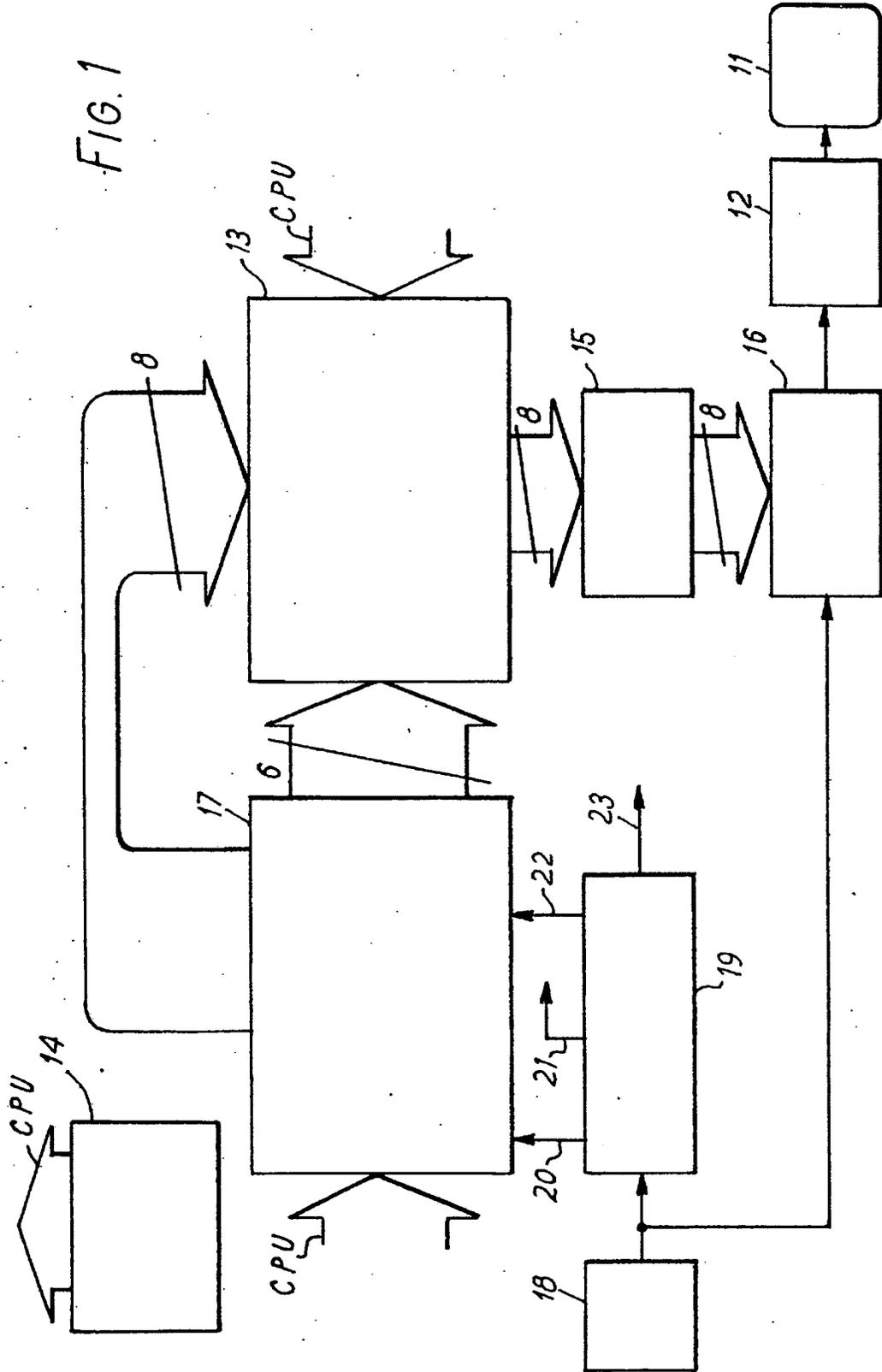
- (21) Application No 7924218
- (22) Date of filing 11 Jul 1979
- (23) Claims filed 11 Jul 1979
- (30) Priority data
- (31) 14400/78
- (32) 12 Apr 1978
- (33) United Kingdom (GB)
- (43) Application published 19 Dec 1979
- (51) INT CL²
G06K 15/20
- (52) Domestic classification
H4T 4A2 4B1
- (56) Documents cited
None
- (58) Field of search
H4T
- (71) Applicants
Data Recall Limited,
Sondes Place, Dorking,
Surrey RH4 3EF
- (72) Inventor
Mark-Eric Jones
- (74) Agents
Reddie & Grose

(54) Video display control apparatus, (57) Video display control apparatus for a visual display device (11, Fig. 1, not shown) employing a television-type raster in a word processor has a display memory (13), a column counter 25 and a row counter 28 adapted to address the display memory. Each location of the display memory has an address comprising a column number and a row number. A clock oscillator (18) and a timing chain (19) produce raster timing signals and column and row timing signals. The count in the column counter 25 tracks the line being scanned, and the count in the row counter tracks successive groups of lines in the raster. The display data output of the display memory controls a character matrix memory (15) acting through a parallel-to-serial converter (16) to cause alphanumeric characters to be displayed in rows by the display device. So that the information display

by the display device can be varied in a convenient manner, the row counter 28 is coupled to the display memory 13 through a random access memory 30 which stores information from a central processor unit (14). This stored information determines which set of sequential row addresses shall be supplied to the display memory as the row counter 28 carries out its counting sequence, and includes an instruction associated with a selected row address which causes a reset signal 31 to be supplied to the column counter 25 so that for this row the characters displayed start at the character stored in the first column of locations in the display memory, the column addresses generated by the column counter 25 being otherwise selectable as any set formed by a predetermined number of consecutive column addresses for alphanumeric character locations in the display memory.



GB2 022 969 A



SPECIFICATION

Video display control apparatus

This invention relates to video display control apparatus for use with a visual display device employing a television-type scanning raster.

Visual display devices are now employed in monitoring or simply displaying information constituting the output of, for example, a computing system, a commercial information disseminating network, or a word processor. At present, such display devices are usually in the form of a cathode ray tube operated with a television-type scanning raster. It is frequently the case that the quantity of data stored in the system supplying the visual display device is greater than the amount that can be displayed simultaneously.

An object of the present invention is to provide control apparatus enabling a visual display device to vary the information display thereby in a convenient manner.

According to the present invention, therefore, there is provided video display control apparatus for use with a visual display device employing a television-type scanning raster, the control apparatus including a display memory, a column counter and a row counter adapted to address the display memory, each of a plurality of locations of the display memory having an address comprising a column number and a row number, timing means for producing raster timing signals and column and row timing signals, the timing means being so coupled to the column counter and the row counter that, in operation, the count in the column counter changes in a manner representative of the scanning of a line of the raster and the count in the row counter changes in a manner representative of the succession of lines in the raster, and means coupled to data output terminals of the display memory for producing display signals representative of display data held in addressed locations of the said plurality of locations, characterised in that the row counter is coupled to the display memory through a random access memory adapted to store a row holding instruction relating to a selected row address and to supply to the column counter a row holding signal such that the column counter in response thereto carries out its column counting or countings for the selected row address through a predetermined series of column numbers, the column counter being adapted to count a predetermined number of column numbers starting from a column number which is selectable except in the presence of the row holding instruction.

Since the count in the column counter changes in a manner representative of the scanning of a line of the raster and the count in the row counter changes in a manner representative of the succession of lines in the raster, and the column and row timing signals are such that the count in the column counter changes faster than the count in the row counter. Although the terms column and row are thus associated with the scanning of

65 a line of the raster and the succession of lines in the raster respectively, the lines of the raster in the display in operation may be so orientated as to run from top to bottom of the display as viewed by a user. Normally, however, the lines will be orientated so as to run from left to right in the display.

Preferred features of the apparatus are defined in the sub-claims appended hereafter.

The invention will now be described in more detail, solely by way of example, with reference to the accompanying drawings, in which:—

Fig. 1 is a block diagram of a word processor embodying the invention; and

Fig. 2 is a block diagram showing in more detail part of the embodiment of Fig. 1.

In the word processor of Fig. 1, a cathode ray display tube 11 receives a video signal from a video output stage 12. Scanning circuitry for the cathode ray display tube 11 is not shown and produces a scanning raster on the screen of the tube 11, the scanning raster being formed by a large number of horizontal lines. The scanning of the raster is similar to that of a television raster except that there is no interlacing of the lines. The lines in the scan making up each frame of the raster are produced in sequence starting at the top of the frame. A display memory 13 stores alphanumeric character codes in a plurality of locations arranged to represent, for example, an array of 128 columns by 64 rows. The character codes are supplied to the display memory 13 by a central processor unit 14 which receives this information from a flexible disc, not shown, or a keyboard, not shown.

Whenever one of the locations containing a character code in the display memory is addressed, the character code is supplied to a character matrix memory 15 which stores a character scan dot code for each possible alphanumeric character. In the present example, each alphanumeric character is formed by a selection of dots from a matrix of 10 by 13 dot positions, each matrix being 13 dots high and 10 dots wide. Consequently, 13 line scans are required to scan each complete character. Thus one row consists of 13 horizontal successive lines of dots, in coded form, supplied by the matrix memory 15 to a parallel-to-serial converter 16 in the form of a 10 bit shift register. The serial output of this converter is supplied to the video output stage 12 which correspondingly supplies video dot signals to the cathode ray display tube 11.

The display memory 13 is addressed by an addressing unit 17 which provides the address for each of the alphanumeric character locations of the display memory in the form of a 6 bit row address combined with an 8 bit column address. In effect, a selected succession of 80 column addresses is supplied 13 times to the display memory 13 during the supplying of each row address to the display memory 13. Consequently, each of the 13 horizontal lines of dots in coded form supplied to the converter 16 consists of 80 groups of dots, each group lying in a respective

column and being a selection of the dots forming the character at the location defined by the respective column and the current row.

Timing signals, in the form of pulses, are generated as follows.

A clock oscillator 18 generates clock pulses at, for example, 50 megahertz. The clock pulses are supplied directly to the shift register constituting the converter 16 and thus the dot rate is set at the frequency of the clock oscillator 18. The clock pulses are also supplied directly to a timing chain 19 which consists of a chain of frequency dividers (not shown). Four outputs 20, 21, 22 and 23 from the timing chain 19 are shown. Streams of pulses at successively lower rates are supplied at these outputs 20 to 23. The highest pulse rate, which is at the output 20, is supplied to the addressing unit 17 to determine the rate at which column addresses are generated. This rate is accordingly the character clock rate and may be, for example, 5 megahertz. The pulses supplied at the output 21 are generated at a rate which is used as the line frequency for the raster of the cathode ray display tube 11. Each pulse at the output 21 is very short and corresponds substantially to a line sync pulse. The rate of the pulses at the output 22 is 1/13th that of the pulses at the output 21. The pulses at the output 22 are supplied to the addressing unit 17 where they serve to determine the row address rate. The rate of the pulses at the output 23 is 1/68th of the rate of the pulses at the output 22. The pulses at the output 23 are accordingly used as frame sync pulses, i.e. the pulses which determine the instants at which rasters on the cathode ray display tube 11 are completed.

The central processor unit 14 supplies to the addressing unit 17 information which determines which succession of 80 of the 128 columns is to be addressed by the addressing unit, and which one of the 64 rows is to serve as the starting row during addressing by the addressing unit. This facility enables the cathode ray display tube 11 to display the information contained in any array of 80 columns by 64 rows selected from the array of 128 columns by 64 rows representing the stored alphanumeric characters in the display memory 13. For example, if the array represented by the locations in the display memory 13 is considered to consist of columns 1 to 128 numbered from the left and rows 1 to 64 numbered from the top, the addressed array may consist of columns 21 to 100 by rows 10 to 64 followed by rows 1 to 9. Furthermore, the information supplied to the addressing unit 17 by the central processor unit 14 can include an instruction for a selected row of the addressed array to consist of the locations in columns 1 to 80 of that row while the other rows consist of the locations in another succession of 80 columns, for example, columns 21 to 100.

The means whereby this latter operation is carried out will now be described with reference to Fig. 2.

In Fig. 2, the addressing unit 17 is shown to consist of a roll left right offset latch 24 which holds the current value of the left hand column to be displayed, this value being supplied to the latch

by the central processor unit, a column counter 25 coupled to the latch 24 to receive therefrom an 8 bit output representing the left hand column value held by the latch 24, and receiving at an input 26 the character rate pulses supplied by the output 20 of the timing chain 19, a buffer 27 coupled to the 8 bit output of the counter 25 and having an 8 bit output at which the column addresses supplied to the display memory 13 appear in operation, a row counter 28 which receives at an input 29 the row rate pulses provided at the output 22 of the timing chain 19, and a random access memory 30 coupled to the row counter 28 to receive therefrom a 6 bit output, and having an 8 bit output of which 6 bits are supplied to the display memory 13 as the row addresses, the 7th bit of the output being supplied to a reset input 31 of the column counter 25 and the 8th bit of the output being supplied to the display memory as a blanking signal to force the main memory to provide no alphanumeric character as output during the active time of the signal on the 8th bit of the output of the random access memory 30. The random access memory 30 also receives an input from the central processor unit which determines the prevailing relationship between the 6-bit output of the row counter 28 and the first 6 bits of the output of the random access memory 30 which are supplied as row addresses to the display memory 13. The input to the random access memory 30 from the central processor unit also determines for each row address generated by the random access memory 30 the accompanying values of the 7th and 8th bits of the output of the random access memory. In particular, the value of the 7th bit for each row address is either high or low, and in response to one of these values, the column counter 25 is reset to zero. The column counter 25 is arranged to count a succession of 112 column numbers starting from the number of the left hand column supplied to it by the latch 25 unless the counter 25 is reset to zero in which case the count of 112 successive column numbers is started at zero. Consequently, in the display on the cathode ray display tube 11, rows of alphanumeric characters are presented which start at the left hand end with the character in the left hand column determined by the value supplied to the counter 25 by the latch 24 when for the row address supplied to the display memory 13 by the random access memory 30 the 7th bit of the output of the random access memory 30 is not such as to reset the column counter 25. However, when the 7th bit of the output of the random access memory 30 accompanying the row address supplied to the display memory 13 is such as to reset the column counter 25, the corresponding row of alphanumeric characters displayed by the cathode ray display tube 11 starts at its left hand end with the character occurring in the first column of locations in the display memory 13 for that row. Line fly-back blanking pulses are supplied to another input 32 of the column counter 25 to set the counter 25 to the start of each cycle of

counting each blanking pulse occurring during the last 32 counts. In the present example, the column counter 25 is capable of counting from 0 to 255. It will be realized that the selection of the left hand column by means of the left hand column number 5 supplied by the latch 24 to the counter 25 enables that area of the array of locations containing alphanumeric characters in the display memory 13 which is to be displayed by the cathode ray display tube 11 to be shifted to the left and to the right. Such shifting is referred to as rolling. The fixing of a particular row to the first 80 columns by the 7th bit of an output from the random access memory 30 enables rows thus selected to be held in the display on the cathode ray display tube 11 while the other rows are rolled to the left or to the right. This facility is particularly useful in the case of rows constituting headings for information appearing in the display.

The row counter 28 is such as to count from 0 to 67 and supplies its count in coded form as the 6 bit output to the random access memory 30. In a manner determined by the instructions received by the random access memory 30 from the central processor unit, the random access memory 30 translates the count of the row counter 28 into an 8 bit output signal in which the first 6 bits constitutes a row address, the 7th bit constitutes the signal to be supplied to the reset input 31 of the column counter, and the 8th bit constitutes a signal to the display memory 13 instructing that memory 13 to either provide the contents of the addressed locations or to provide a blank output signal.

The counting operation carried out by the row counter 28 is synchronised with the raster of the cathode ray display tube 11 so that the counts 64, 65, 66, and 67 occur during the frame fly-back blanking time. This locking of the counting cycle of the counter 28 to the raster timing ensures that rows of characters are automatically placed in the desired positions in the displayed array.

The random access memory 30 may be a Motorola MCM 6810AL which has a capacity of a 128 times 8 bits. The display memory 13 may be formed of 32 Texas Instruments TMS4044—15, each being a 4K by 1 bit static random access memory. The character matrix memory 15 may be formed of 8 Texas Instruments TMS4044—15. Where the random access memory 30 is a Motorola MCM 6810AL, the 6 bit input from the row counter 28 is multiplexed with the input which the random access unit 30 receives from the central processor unit.

55 CLAIMS

1. Video display control apparatus for use with a visual display device employing a television-type scanning raster, the control apparatus including a display memory, a column counter and a row counter adapted to address the display memory,

each of a plurality of locations of the display memory having an address comprising a column number and a row number, timing means for producing raster timing signals and column and row timing signals, the timing means being so coupled to the column counter and the row counter that, in operation, the count in the column counter changes in a manner representative of the scanning of a line of the raster and the count in the row counter changes in a manner representative of the succession of lines in the raster, and means coupled to data output terminals of the display memory for producing display signal representative of display data held in addressed locations of the said plurality of locations, characterised in that the row counter is coupled to the display memory through a random access memory adapted to store a row holding instruction relating to a selected row address and to supply to the column counter a row holding signal such that the column counter in response thereto carries out its column counting or countings for the selected row address through a predetermined series of column numbers, the column counter being adapted to count a predetermined number of column numbers starting from a column number which is selectable except in the presence of the row holding instruction.

2. Apparatus according to claim 1, wherein a latch for storing a selected column number is coupled to the column counter, and the column counter is adapted to effect counting of a predetermined number of column numbers starting from the column number stored in the latch except in the presence of the row holding instruction.

3. Apparatus according to claim 1 or 2, characterised in that the column counter has a reset input terminal, the random access memory is so coupled to the column counter as to supply row holding instructions to the reset input terminal, and the column counter is such as to reset to the count zero whenever a row holding instruction is present at the reset input terminal.

4. Apparatus according to claim 3, characterised in that the random access memory is adapted to encode the count in the row counter as a different count related thereto by a constant which is selectable,

5. Apparatus according to claim 4, wherein the said locations of the display memory are filled by a central processor unit which is arranged to supply the column number to be stored to the said latch, and to supply the instructions to the random access memory which determine the said constant and determine the said selected row address.

6. Video display control apparatus substantially as described herein before with reference to the accompanying drawings.

(12) UK Patent Application (19) GB (11) 2 354 102 (13) A

(43) Date of A Publication 14.03.2001

(21) Application No 9921227.6

(22) Date of Filing 08.09.1999

(71) Applicant(s)
Barron McCann Limited
 (Incorporated in the United Kingdom)
 BeMac House, Fifth Avenue, LETCHWORTH,
 Hertfordshire, SG6 2HF, United Kingdom

(72) Inventor(s)
 Peter Alderson
 Robert Andrew Edge

(74) Agent and/or Address for Service
 Williams, Powell & Associates
 4 St Paul's Churchyard, LONDON, EC4M 8AY,
 United Kingdom

(51) INT CL⁷
 G07F 7/10 , G06F 17/60

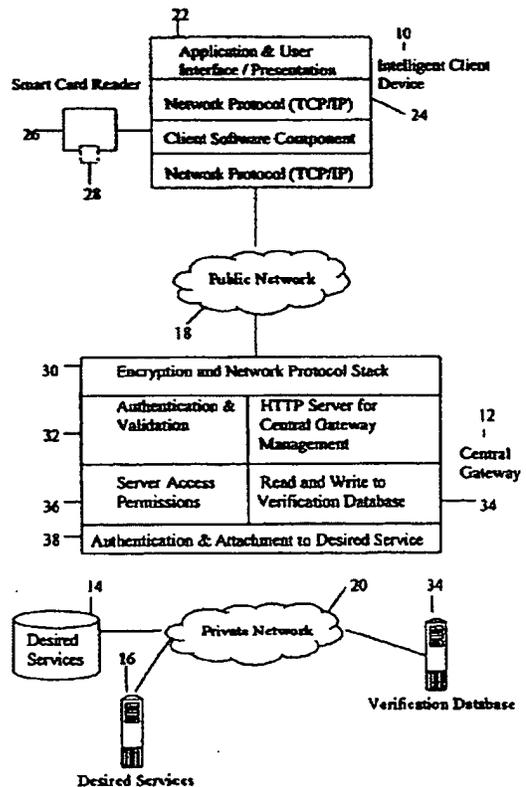
(52) UK CL (Edition S)
 G4V VAK

(56) Documents Cited
 EP 0813175 A2 WO 98/32260 A1 WO 97/50207 A1
 WO 97/29416 A2 US 5809143 A

(58) Field of Search
 UK CL (Edition R) G4V VAK , H4P PDCSA
 INT CL⁷ G06F 17/60 , G07F 7/10
 Online: WPI, EPODOC, JAPIO

(54) Abstract Title
System for communicating over a public network

(57) A system for communicating with a remote service over a public network 18, such as the Internet, includes a client device 10 with a memory card 28 or the like, a card reader 26 and a public network communication device such as a personal computer or television, and a processor unit, such as a central gateway 12, which is located remotely from the client device. The memory card includes user details which are transmitted by the client device to the processor unit, and may be encrypted. The card reader may activate communication with the processor unit upon insertion of the memory card, which may be a smart card or magnetic card. The processor unit may determine which of a plurality of services 14,16 a user is authorised to access. The system provides for secure communication without burdening the user with encryption or authorisation tasks.



GB 2 354 102 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

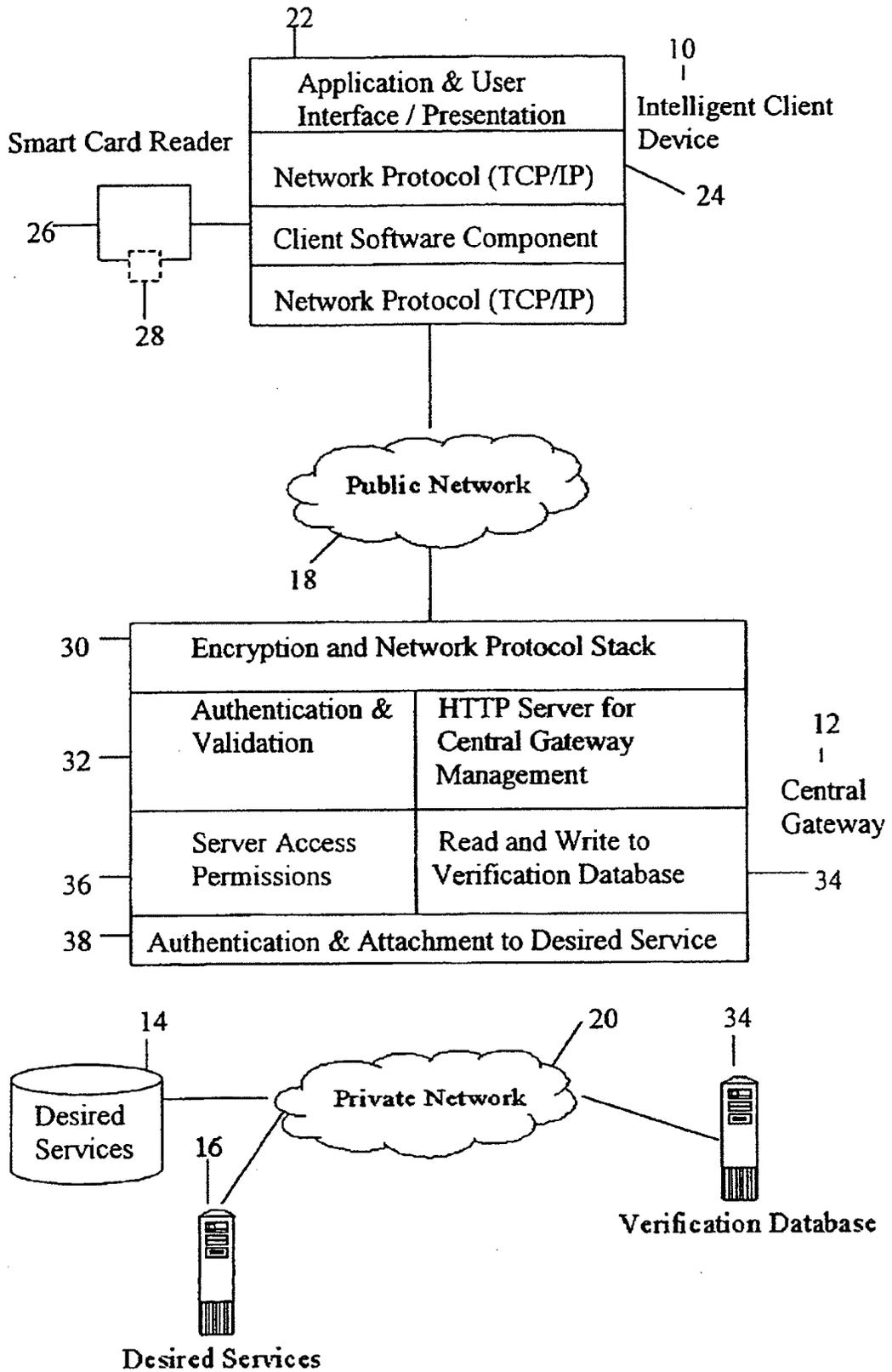


Fig 1

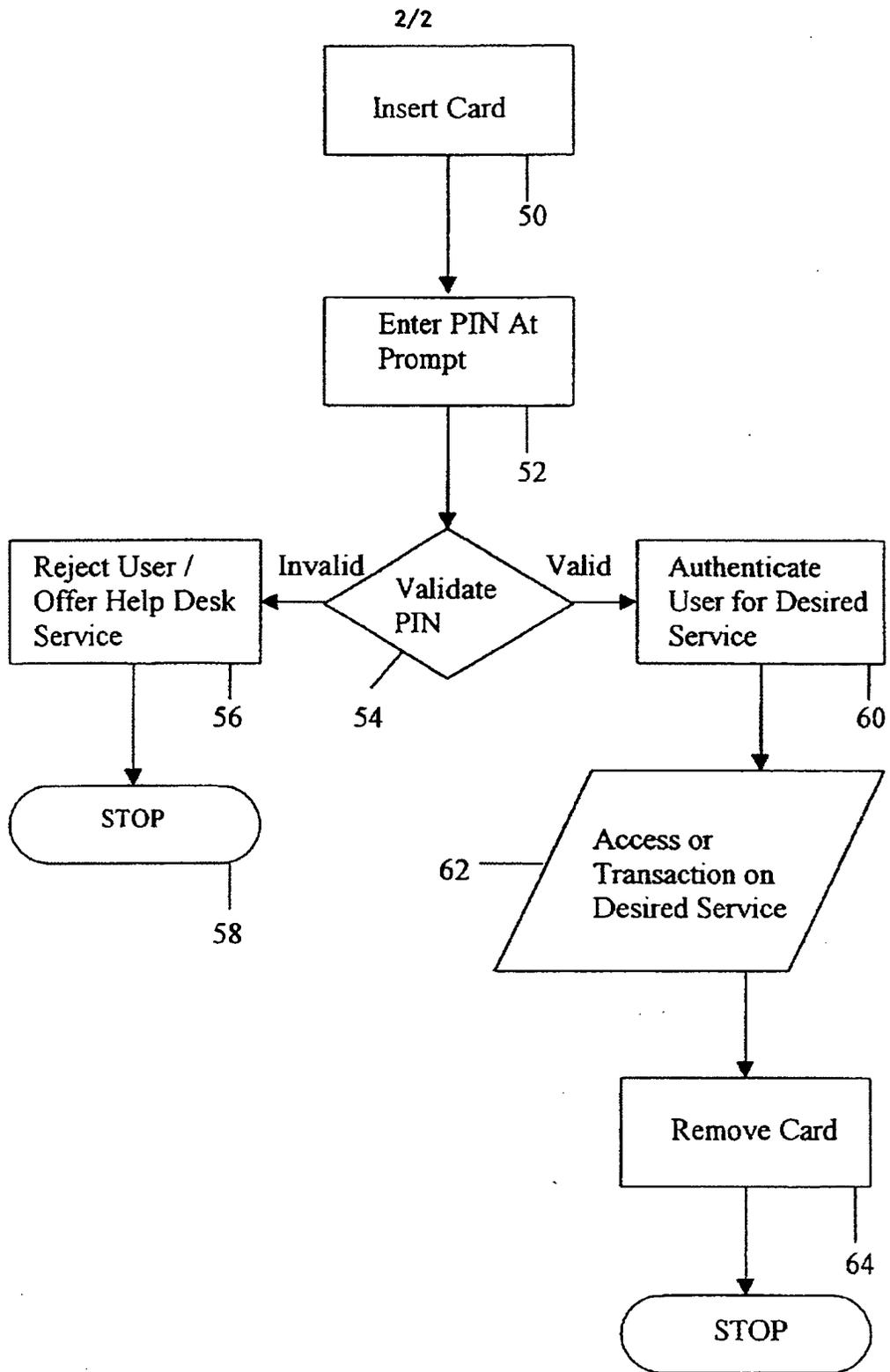


Fig 2

SECURITY SYSTEM

The present invention relates to a security system, for use for example in accessing remote services such as on the Internet.

5 With the advent of modern technology, a growing number of transactions are being carried out by the user across insecure networks. These can be, for example, transactions involving confidential data and money for payment or investment. With such transactions there are problems with security, fraud and so on. Various security systems have been devised, such as use of personal identification numbers, encryption of
10 transmissions. While these systems usually work well for the particular environment for which they have been designed, they can be a nuisance to use and can be difficult or expensive to implement for a new service provider.

Systems have also been developed for Internet use. These systems concentrate on
15 authentication of the user and then, once this has been established, provide for un-encrypted connection to the service. When particular transactions are undertaken, the service determines whether encryption is necessary, for example to secure credit card details. Other solutions require entry of credit card details for each transaction. These systems inevitably must provide a balance between security and user convenience as the
20 encryption mechanisms used cause additional work for and complication to the user.

The present invention seeks to provide an improved security system.

According to an aspect of the present invention, there is provided a security system for
25 communicating with a remote service over a public network including a user card or other memory device, a user located card or memory device reader, a user located public network communication device and a processor unit located remotely from the user located public network communication device, wherein the user card includes user details and the user located public network communication device is operable to transmit the
30 user details to the processor unit.

Advantageously, the processor unit is operable to carry out encryption between it and the user and to provide to the user a transparent path to the service. Thus, the user need not be aware of any security steps taken or any encryption system used, this being carried out by the card reader and the processor unit or central gateway.

5

The card may be any suitable device which can store user information and, preferably, encryption data. The card, can for example be a smart card, a magnetic card such as a credit/debit card or store loyalty card or any other suitable device. In addition to the card, the user may be required to input a secret identification code, such as an
10 identification number.

In the preferred embodiment, the system provides for the user to insert the card into his/her card reader and to initiate the connection to the processor unit or central gateway. Once the connection is made, the processor unit obtains the relevant data from the card
15 and upon verification by the identification code, allows the user access to the authorised service without any intermediate tasks, such as requirements to encrypt or decrypt transmitted data, to provide other user details and, where appropriate account or payment details. Thus, as with the preferred embodiment, all communications between the
processor unit and the user can be encrypted, without the user necessarily being aware of
20 or involved in this encryption. The communication between the user and the processor unit can therefore be totally secure yet without user inconvenience.

Advantageously, communications between the service and the processor unit, which are preferably carried out via a secure link, need not be encrypted.

25

The splitting of the encryption from the service results in being able to provide a dedicated encryption device, the processor unit, which can therefore be designed to maximise encrypted communication efficiency. Typically, encryption of all communications from the service unit is not practicable because the service unit is not
30 designed for such a task and even if it were it would result in a loss of efficiency in providing the service itself.

In the preferred embodiment, the processor unit is also able to determine which of a plurality of services the user is authorised to access and/or the level of access such as spending limit, and to control access to the service or relevant service on this basis. It
5 can also or alternatively undertake transactions against an account identified by the card.

An embodiment of the present invention is described below, by way of example only, with reference to the accompanying drawings, in which:

10 Figure 1 is a schematic diagram of an embodiment of security system coupled to a processor unit or central gateway and a service; and

Figure 2 is a flow chart of an example of validation routine for use with the system of Figure 1.

15

Referring to Figure 1, the embodiment of security system shown is designed for communications through the Internet or a similar public network.

The system includes an intelligence client device 10, which may be a personal computer, television, or any other suitable device which can communicate with a remote system. A
20 processor unit, in this example a central gateway 12 is coupled between the client device 10 and one or more service units 14.

Communication between the client device 10 and the central gateway 12 is, in this
25 embodiment, via a public network 18 such as the Internet. Communication between the central gateway 12 and the service units 14, 16 is, on the other hand, via a private network 20 which cannot be accessed by the public.

The client device 10 is provided with an application and user interface 22; which can be
30 the usual computer devices such as monitor, keyboard and software in the case that it is a personal computer; the screen and a suitable keyboard or keypad in the case that the

device 10 is a television or any other suitable device. The device 10 could also be a portable telephone with suitable display and keypad.

5 The device 10 also includes suitable network protocol 24 for allowing communication to the gateway 12 through the chosen network 18 or other public transmission medium.

The device 10 also includes a card reader 26 designed for reading the card-type chosen for the system and a card 28 which is specific to that user. The card 28 could be a smart card or magnetic card of the types well known or any other portable memory device. It
10 is envisaged that the card 28 could have other functions in addition to the security function for this system, for example it could also be a credit/debit card, store loyalty card and the like.

The card 28 has stored thereon one or more user identifiers, one or more encryption keys
15 and the desired service information, that is details of the service to which the user wants access. His/her level of authorisation in the service and so on will be determined by the central gateway 12.

The card reader 26 is designed, in the preferred embodiment, to be able to detect the
20 insertion of the card 28 thereinto and in response to such insertion to commence immediately communication with the gateway 12 via the client device 10.

The central gateway 12 includes an encryption and network protocol stack 30 designed to allow communication via the chosen public network 18 and to provide encryption of all
25 communications between itself and the client device 10. It also includes an authentication and validation unit 32 for authenticating the client data from the client card 28. The authentication and validation unit 32 is coupled to a verification database 34 of the gateway 12 in which is stored the identification data of all the users registered for the services 14,16. The database 34 may be provided either within the gateway 12 or in a
30 remote database 34' accesses through secure network 20.

The authentication and validation unit 32 is also coupled to server access permission unit 36 designed to control the type of access to the service units 14,16 in dependence upon the user's authority.

5 Also provided in the gateway 12 are a typical HTTP server for management of the gateway 12 and an authentication and attachment unit 38 for communicating with the desired services 14,16 and with any remote verification database 34'.

The central gateway 12 is designed specifically for encrypting all communications over
10 the public network 18 and for carrying out the authentication procedure.

The operation of the this embodiment will now be described with reference to Figure 2.

Insertion 50 of the card 28 into the card reader 26 prompts the card reader 26 to
15 commence automatically the connection to the gateway 12. For this purpose, card reader 26 activates a software component in the device 10 to establish a communication link with the gateway 12 on the basis of information stored on the card 28 about the location on the Internet and access details of the gateway 12.

20 When a connection with the gateway 12 is established, the gateway 12 requests the user's personal identification code which is then inputted 52 at a suitable prompt on the user interface 22.

Validation 54 of the user's details and identification code is carried out either internally
25 of the gateway 12, by the units 32 and 34, or externally at the verification database 34'.

If the gateway 12 determines 54 that the user's identification code is invalid, the user is rejected 56 and the connection is cut 58. On the other hand, if it is determined 54 the user's identification code is valid, the gateway 12 determines 60 the desired service 14,
30 16 and level of service to be provided and connects 62 to the desired service unit 14, 16.

During the connection to the desired service 14, 16, all data transfers between the gateway 12 and user device 10 are encrypted on the basis of the encryption keys on the user's card 28 and within verification database 34, while all data transfers between the gateway 12 and the service units 14, 16 through the private network 20 are not encrypted
5 for ease of access and for increased efficiency. In practice, the user will not be aware of the encryption between him/her and the gateway 12 as this will be carried out as a background task. Moreover, the user will not need to re-confirm his/her identity or financial details as these will be provided by the card 28 or gateway 12.

10 The gateway 12, in some embodiments, records the activities of the client, such as transaction details, either within the gateway 12 or in a remote memory accessed via a private network.

Disconnection from the services 14, 16 is, in this embodiment, effected simply by
15 removing 64 the card 28 from the card reader 26.

Thus, connection is made by a simple two step process of inserting the card 28 into the reader 26 and entering the user identification code and disconnection is effected by removing the card 28 from the card reader 26. The user is not involved in any other
20 authentication or encryption process and need not re-enter personal details.

This system can be used for any remote service, including business to consumer (in which case the card could be designed also to function as a store or credit card), business to business (for example for transactions on account) and for internal networking (where
25 the activity of staff, for example, needs to be secured).

It will be apparent from the above that the system can provide simple but absolutely secure access to a remote service. Moreover, by identifying the user to the desired service, user access can be customised. By removing the need for entry of account
30 details, transactions into the desired service become quicker and less risky for the user's perspective.

Performance of the services can also be enhanced by carrying out the encryption tasks within the gateway rather than in the service units.

- 5 In addition, the service company can establish a relationship with the user by providing the user with the card and, possibly, also with the card reader.

It will be apparent that the card 28 and card reader 26 could be configured to communicate with a plurality of separate gateways 12.

10

CLAIMS

1. A security system for communicating with a remote service over a public network including a user card or other memory device, a user located card or memory device reader, a user located public network communication device and a processor unit located remotely from the user located public network communication device, wherein the user card includes user details and the user located public network communication device is operable to transmit the user details to the processor unit.
2. A security system according to claim 1, wherein the processor unit is operable to carry out encryption between itself and the user.
3. A security system according to claim 1 or 2, wherein the card has stored thereon user information and, preferably, encryption data.
4. A security system according to claim 3, wherein the card is a smart card, a magnetic card or any other suitable device.
5. A security system according to any preceding claim, wherein the card reader is operable to activate communication with the remote processor means upon insertion of a card thereinto.
6. A security system according to any preceding claim, wherein the processor unit is operable to encrypt substantially all communications between the user and itself.
7. A security system according to any preceding claim, wherein the processor unit is operable to determine which of a plurality of services a user is authenticated onto the desired service.

30

8. A security system substantially as hereinbefore described with reference to and as illustrated in the accompanying drawings.



Application No: GB 9921227.6
Claims searched: All

Examiner: Michael Logan
Date of search: 20 January 2000

**Patents Act 1977
Search Report under Section 17**

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:
UK CI (Ed.R): G4V (VAK); H4P (PDCSA)
Int CI (Ed.7): G06F 17/60; G07F 7/10
Other: Online: WPI, EPODOC, JAPIO

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X	EP 0813175 A2 (NCR INTERNATIONAL) whole document relevant	1-6
X	WO 98/32260 A1 (COMMONWEALTH BANK OF AUSTRALIA) see page 2 and fig 1	1-6
X	WO 97/50207 A1 (TELIA AB) see page 9, lines 1-24	1-6
X	WO 97/29416 A2 (INTEGRATED TECHNOLOGIES OF AMERICA) see especially page 7, line 5 - page 8, line 16	1-7
X	US 5809143 (HUGHES) see for example column 10, lines 35-43	1-6

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

16) Family number: 12389386 (JP11031130 A2)

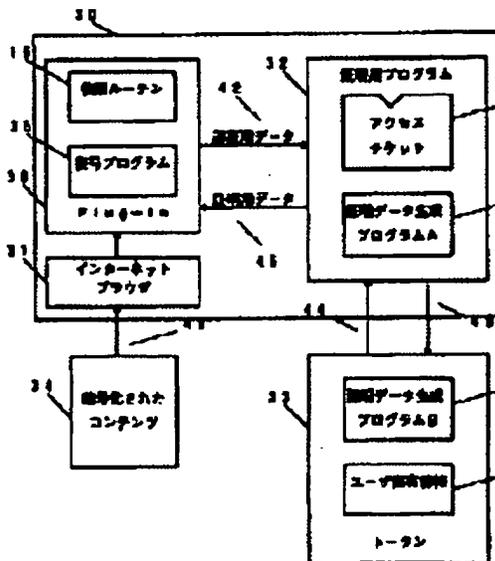
full-text | status | citations | < | > | ^ |

Title: SERVICE PROVIDING DEVICE
 Priority: JP19970184866 19970710
 Priority Map

Family:	Publication number	Publication date	Application number	Application date	Link
Family Explorer	JP11031130 A2	19990202	JP19970184866	19970710	

Assignee(s): FUJII XEROX CO LTD
 Inventor(s): KOJIMA SHUNICHI ; KONO KENJI ; NAKAGAKI JUHEI
 International G06F15/00 G09C1/00 H04L9/32 (Advanced/Invention);
 class (IPC 8): G06F15/00 G09C1/00 H04L9/32 (Core/Invention)
 International G06F15/00 G09C1/00 H04L9/32
 class (IPC 1-7):

Abstract:
 Source: JP11031130A2 PROBLEM TO BE SOLVED: To provide the utilization of service only to a user who has a legal right, minimizing the burden on the user and a service provider. SOLUTION: When a plug-in 38 of an internet browser 31 is started, a verification program 15 in the plug-in 38 is started, communicates with a program 32 for certification and performs user authentication. A certification data generation program A36 of the program 32 cooperates with a certification data generation program B37 in a token 33, calculates based on a user inherent information 16 and an access ticket 13 and communicates with the program 15 in the plug-in 38 based on the calculation. As the result of the communication, the success of authentication by the program 15 is limited to only when the three of the user inherent information, the access ticket and enciphered contents correctly correspond with one another.



17) Family number: 12393236 (JP11032037 |  |  | full-text | status | citations | < | > | ^ |  | 

Title: CERTIFICATION DATA GENERATING DEVICE

Priority: JP19970188801 19970714
Priority Map

Family:	Publication number	Publication date	Application number	Application date	Link
Family Explorer	JP11032037 A2	19990202	JP19970188801	19970714	
	JP3641909 B2	20050427	JP19970188801	19970714	

Assignee(s): FUJI XEROX CO LTD

Inventor(s): NAKAGAKI JUHEI ; SHIN YOSHIHIRO

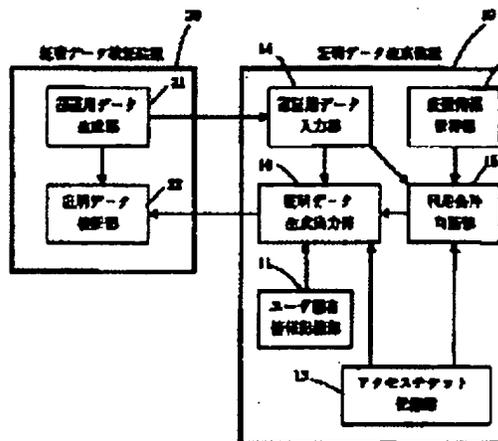
International class (IPC 8): G06F15/00 G06F9/06 G09C1/00 H04L9/32 (Advanced/Invention);
 G06F15/00 G06F9/06 G09C1/00 H04L9/32 (Core/Invention)

International class (IPC 1-7): G06F15/00 G06F9/06 G09C1/00 H04L9/32

Abstract:

Source: JP11032037A2 PROBLEM TO BE SOLVED: To pre-pay access qualification to purchase or rent without imposing any surplus load on a certification data generating device side. SOLUTION: A pre-paid purchase ticket T_2 is stored in an access ticket storing part 13.

Next, (T_1', n_2) is inputted to a certification data-imputting part 14. A use condition judging part 15 extracts a corresponding access ticket (t_2, L_2, n_2) , checks whether or not a use condition L_2 is fulfilled, and reduces frequency information V , when the use condition is fulfilled. A certification data generating and outputting part 16 calculates certification data R by using auxiliary certification decision $(t)_2$ and the use condition L_2 extracted by the use condition decision part 15 and (du) read from a user specific information storing part 11, and outputs T_1 . A user performs access to a program in a purchase state or a rent state by using the T_1 .



12) Family number: 13081077 (JP11205306 A2)

full-text | status | citations | < | > | ^ |

Title: AUTHENTICATION SYSTEM AND AUTHENTICATION METHOD

Priority: JP19980006267 19980116
Priority Map

Family:	Publication number	Publication date	Application number	Application date	Link
<u>Family Explorer</u>	JP11205306 A2	19990730	JP19980006267	19980116	

Assignee(s): FUJI XEROX CO LTD

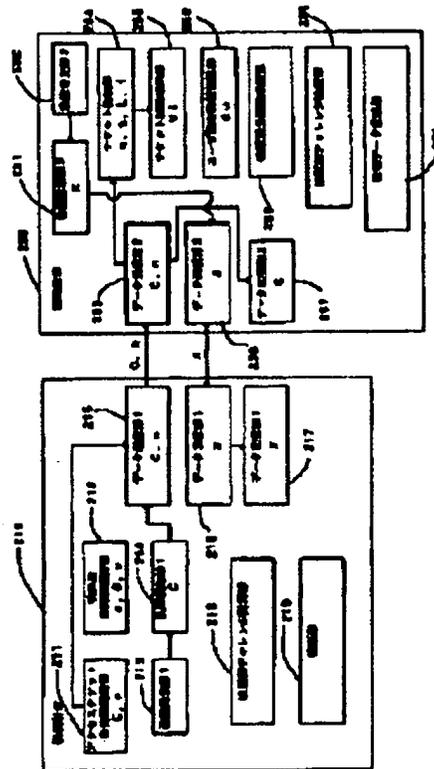
Inventor(s): KOJIMA SHUNICHI ; KONO KENJI ; TAGUCHI MASAHIRO ; TERA0 TARO

International class (IPC 8): G09C1/00 H04L9/32 (Advanced/Invention);
 G09C1/00 H04L9/32 (Core/Invention)

International class (IPC 1-7): G09C1/00 H04L9/32

Abstract:

Source: JP11205306A2
PROBLEM TO BE SOLVED: To provide a system and method that realize diversified services by using an access ticket generated from characteristics information not belonging to a person and information specific to the user, as for the authentication system that authenticates legality of the user.
SOLUTION: The authentication device 210 sends authentication data and a ticket identifier to an authentication device 250, the authentication device 250 sends the authentication data to the authentication device 210, which calculates an authentication challenge (p) based on a ticket attribute revision request (μ) and an authentication device authentication data (x). The authentication device 250 receives the p and the (β eps1, v) to authenticate an authentication device open key based on input data and an authentication device open key identifier (v'), to authenticate the authentication device challenge and to revise contents of a ticket attribute record (Vi) depending on the ticket attribute revision request (μ). Furthermore, an authentication data generating section calculates a response (R) and the authentication device authenticates the legality of the response (R).



11) Family number: 13107360 (JP11215121 A2)

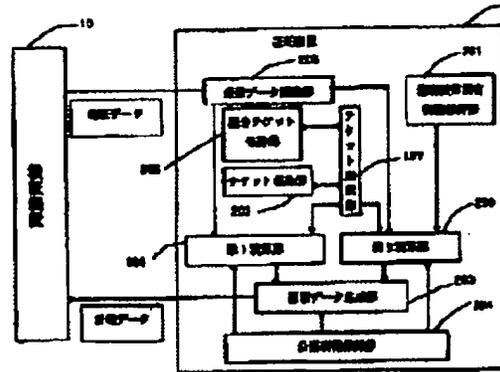
full-text | status | citations | > |

Title: DEVICE AND METHOD FOR AUTHENTICATION
Priority: JP19980016710 19980129
Priority Map

Family:	Publication number	Publication date	Application number	Application date	Link
<u>Family Explorer</u>	JP11215121 A2	19990806	JP19980016710	19980129	
	JP3791169 B2	20060628	JP19980016710	19980129	

Assignee(s): FUJI XEROX CO LTD
Inventor(s): KIKO KENICHIROU
International class (IPC 6): G09C1/00 H04L9/32 (Advanced/Invention);
class (IPC 8): G09C1/00 H04L9/32 (Core/Invention)
International class (IPC 1-7): G09C1/00 H04L9/32

Abstract:
Source: JP11215121A2 **PROBLEM TO BE SOLVED:** To perform composite authentication by using the combination of different kinds of issued tickets.
SOLUTION: The ticket holding section 202 of a certifying device 20 holds a ticket indicating the specific right of a user while a composite ticket holding section 206 holds a composite ticket for certifying that the user holds a plurality of other effective tickets. A certifying data generating section 203 certifies the presence of a compositely designated right by generating certifying data through executing a prescribed operation by the use of a prescribed access ticket, a composite ticket, and inherent information of the certifying device to authentication information sent from a verifying device 10.



8) Family number: 14153892 (JP2000215165 A2)

full-text | status | citations | < | > | ^ |

Title: METHOD AND DEVICE FOR INFORMATION ACCESS CONTROL AND RECORD MEDIUM RECORDING INFORMATION ACCESS CONTROL PROGRAM

Priority: JP19990017401 19990126
[Priority Map](#)

Family:	Publication number	Publication date	Application number	Application date	Link
Family Explorer	JP2000215165 A2	20000804	JP19990017401	19990126	

Assignee(s): NIPPON TELEGRAPH AND TELEPHONE (std):

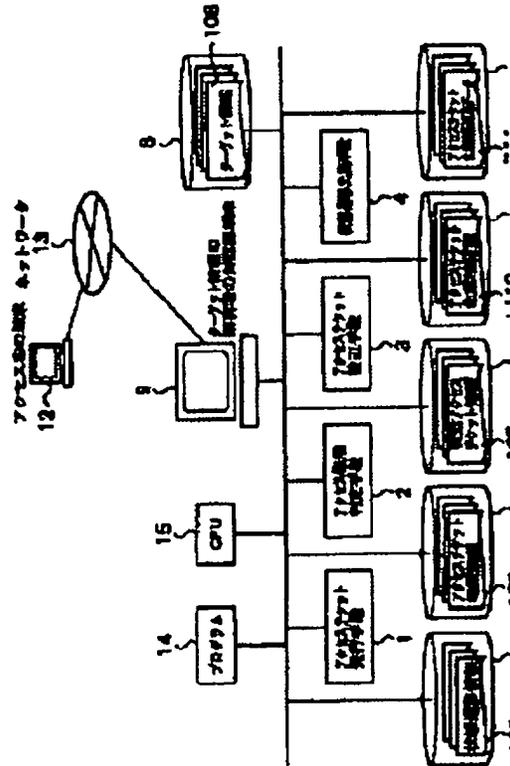
Inventor(s): OHARA YASUHIRO ; OSHIMA YOSHITO

International class (IPC 8): G06F12/14 G06F15/00 G09C1/00 H04L9/32 (Advanced/Invention); G06F12/14 G06F15/00 G09C1/00 H04L9/32 (Core/Invention)

International class (IPC 1-7): G06F12/14 G06F15/00 G06F17/60 G09C1/00 H04L9/32

Abstract:

Source: JP2000215165A2 **PROBLEM TO BE SOLVED:** To provide the method and device for information access control which can easily change the access authority to be allowed to an accessing person in response to the change of situation of a transaction and also to provide a recording medium which records an information access control program. **SOLUTION:** An access ticket issuing means 1 issues the access tickets to every accessing person and these tickets prescribe the access authority to the target information for each of plural types and states. Receiving an access request from an accessing person, the means 1 reads the request and the access authority corresponding to the type and state of an inputted access ticket out of an access ticket authority information storing means 6 and decides to permit or not permit the access request based on the access authority. When a state transition request is received from the accessing person, the transition destination state is read out of a state transition information storing means 5 based on the type and state of the access ticket that is inputted together with the state transition request. Based on the transition destination state, the change of the access ticket is updated.



2) Family number: 33529418 (JP2005218143 A2)
 extended family

text | status | citations | < | > | ^ | ☐ | ☐

Title: ENCRYPTION DEVICE USED IN A CONDITIONAL ACCESS SYSTEM
Priority: US19970054575P 19970801
Priority Map

Family:	Publication number	Publication date	Application number	Application date	Link
<u>Family Explorer</u>	JP2005218143 A2	20050811	JP20050120426	20050418	
	WO9907150 A1	19990211	WO1998US16145	19980731	

Assignee(s): SCIENTIFIC ATLANTA
 (std):

Assignee(s): SCIENTIFIC ATLANTA INC

Inventor(s): PALGON MICHAEL S ; PINDER HOWARD G
 (std):

Designated states: AL AM AT AU AZ BA BB BE BF BG BJ BR BY CA CF CG CH CI CM CN CU CY CZ DE DK EE ES FI GA GB GE GH GM GN GR GW HR HU ID IE IL IS IT JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK ML MN MR MW MX NE NL NO NZ PL PT RO RU SD SE SG SI SK SL SN SZ TD TG TJ TR TT UA UG UZ VN YU ZW

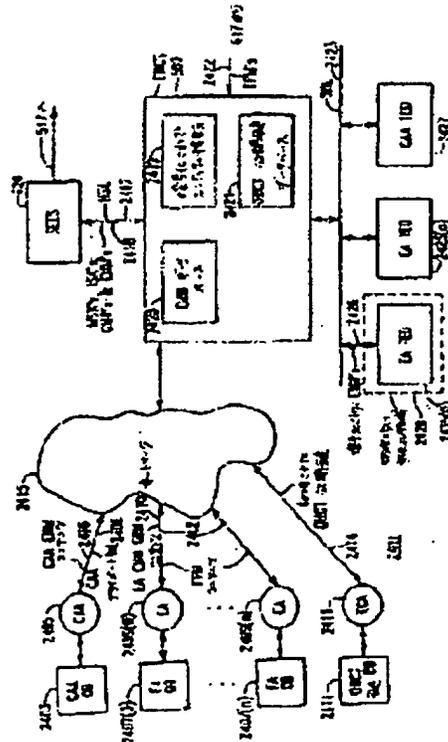
International class (IPC 8): G09C1/00 H04L9/08 H04L9/10 H04N7/10 H04N7/16 H04N7/167 (Advanced/Invention); G09C1/00 H04L9/08 H04L9/10 H04N7/10 H04N7/16 H04N7/167 (Core/Invention)

International class (IPC 1-7): H04L9/10 H04N7/16 H04N7/167

European class: H04N7/167D H04N7/16E2

Cited documents: WO9529560, US5787172, US5592552, US5400401, US5341425, EP0752786.

Abstract:
 Source: JP2005218143A2
PROBLEM TO BE SOLVED: To provide a cable television system providing conditional access to a service. **SOLUTION:** The cable television system includes a headend from which service "instances", or programs, are broadcast and a plurality of set top units for receiving the instances and selectively decrypting these instances for display to system subscribers. The service instances are encrypted, by using public and/or private keys provided by service providers or central authorization agents. Keys, used by the set tops for selective decryption may also be public or private in nature, and these keys may be reassigned at different times, to provide a cable television system in which the anxiety for violation actions is minimized. COPYRIGHT: (C)2005, JPO&NCIPI<



4) Family number: 33529421 (JP2005253109 A2)
 extended family

text | status | citations | < | > | ^ | □ | ☒ | full-

Title: CONDITIONAL ACCESS SYSTEM
Priority: US19970054575P 19970801 US19980126921 19980731
Priority Map

Family:	Publication number	Publication date	Application number	Application date	Link
<u>Family Explorer</u>	JP2005253109 A2	20050915	JP20050120425	20050418	
	WO9909743 A2	19990225	WO1998US16079	19980731	
	WO9909743 A3	19990527	WO1998US16079	19980731	

Assignee(s): SCIENTIFIC ATLANTA
 (std):

Assignee(s): SCIENTIFIC ATLANTA INC

Inventor(s): AKINS GLENDON L III ; PALGON MICHAEL S ; PINDER HOWARD G ; WASILEWSKI ANTHONY J
 (std):

Inventor(s): AKINS GLENDON L

Designated states: AL AM AT AU AZ BA BB BE BF BG B) BR BY CA CF CG CH CI CM CN CU CY CZ DE DK EE ES FI F
 GA GB GE GH GM GN GR GW HR HU ID IE IL IS IT JP KE KG KP KR KZ LC LK LR LS LT LU LV M
 MD MG MK ML MN MR MW MX NE NL NO NZ PL PT RO RU SD SE SG SI SK SL SN SZ TD TG TJ
 TR TT UA UG UZ VN YU ZW

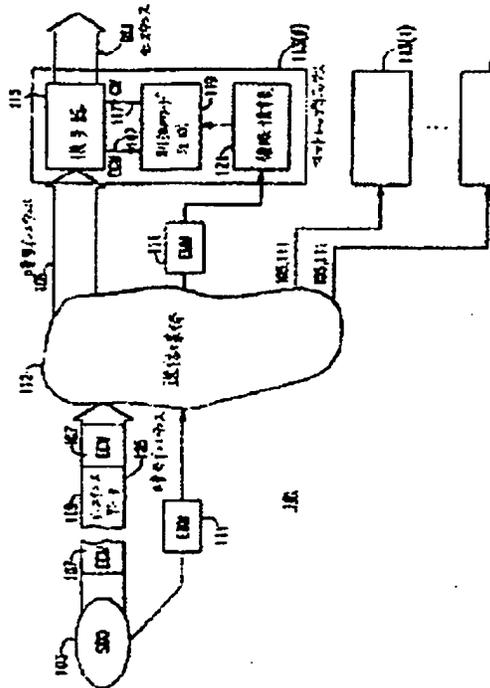
International class (IPC 8): H04H1/00 H04L9/08 H04N5/00 H04N7/16 H04N7/167 H04N7/173 (Advanced/Invention);
 H04H1/00 H04L9/08 H04N5/00 H04N7/16 H04N7/167 H04N7/173 (Core/Invention)

International class (IPC 1-7): H04L9/08 H04N7/167

European class: H04N5/00M4 H04N7/167D H04N7/16E2

Cited documents: WO9704553, US5381481, US5029207, US4887296, US4864615, US4736422, US4613901,

Abstract:
 Source: JP2005253109A2 PROBLEM TO BE SOLVED: To provide a cable television system which provides conditional access to services. SOLUTION: This cable television system includes a headend from which service "instances" or programs are broadcast and a plurality of set top units for receiving the instances and selectively decrypting the instances for display to system subscribers. The service instances are encrypted using public keys and/or private keys provided by service providers or central authorization agents. Keys used by the set tops for a selective decryption may also be public or private in nature, and such keys may be reassigned at different times to provide a cable television system in which piracy concerns are minimized. COPYRIGHT: (C)2005, JPO&NCIPI<



1/9/1
DIALOG(R)File 347: JAPIO
(c) 2008 JPO & JAPIO. All rights reserved.

08787202 ****Image available****
CRYPTOGRAPHIC KEY SYSTEM

Pub. No.: 2006-180562 [JP 2006180562 A]
Published: July 06, 2006 (20060706)
Inventor: SAITO MAKOTO
MOMIKI JUNICHI
Applicant: INTARSIA SOFTWARE LLC
Application No.: 2006-082675 [JP 200682675]
Division of 07-346095 [JP 95346095]
Filed: March 24, 2006 (20060324)
Priority: 06-309292 [JP 94309292], JP (Japan), December 13, 1994 (19941213)

International Patent Class (v8 + Attributes)
IPC + Level Value Position Status Version Action Source Office:

H04L-0009/08 A I F B 20060101 20060609 H JP

ABSTRACT

PROBLEM TO BE SOLVED: To provide a concrete structure for applying a cryptographic key system to a television system, a database system or an electronic commercial transaction system or the like.

SOLUTION: This system consists of a broadcasting station, a database, a receiving apparatus, a data communications apparatus and a user terminal. As a cryptographic key system, a secret-key system, a public-key system, and a digital signature system are used. The keys used in the system are either encrypted, or remain unencrypted to be supplied by broadcasting. The system is effective in preventing the unauthorized use of the database system, managing copyrights, and in pay-per-view systems and video-on-demand systems. Further, the system is effective in realizing an electronic market which uses an electronic data information system.

COPYRIGHT: (C)2006,JPO&NCIPI

55) Family number: 10272458 (JP5168039 A2)

full-text | status | citations | < | > | ^ |

Title: RECORDING ENCODE METHOD FOR HIGH FIDELITY TELEVISION SIGNAL

Priority: JP19910352059 19911213
 Priority Map

Family:	Publication number	Publication date	Application number	Application date	Link
Family Explorer	JP3185806 B2	20010711	JP19910352059	19911213	
	JP5168039 A2	19930702	JP19910352059	19911213	

Assignee(s): SONY CORP

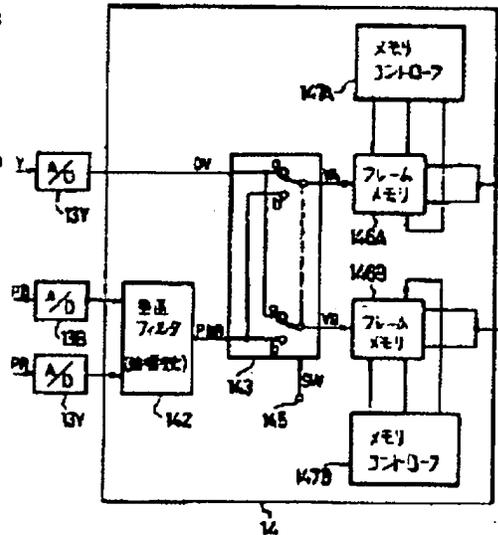
Inventor(s): ISHIMARU HIROYOSHI

International H04N11/22 H04N5/907 H04N9/80 H04N9/81 (Advanced/Invention);
 class (IPC 8): H04N11/06 H04N5/907 H04N9/80 H04N9/81 (Core/Invention)

International H04N11/22 H04N5/907 H04N9/80 H04N9/81
 class (IPC 1-7):

Abstract:

Source: JP5168039A2 PURPOSE: To encode a unit signal (TDM signal) for recording from a high fidelity television signal by controlling reading of plural output ports while using a serial access memory equipped with the plural output ports. CONSTITUTION: Memories 146A and 146B are serial access and two output ports are respectively provided in each memory. Then, write of input data VA and VB is controlled by memory controllers 147A and 147B, and reading of data from the respective output ports is independently controlled. Namely, TDM signals are written in memories 146A and 146B in the order of a luminance signal and a chrominance signal. In the case of reading, the same data are read from two output ports while deviating read timing, color difference signal data are extracted from the preceding output port, luminance signal data are extracted from the other output port, both data are synthesized and therefore, the required TDM signals are obtained.

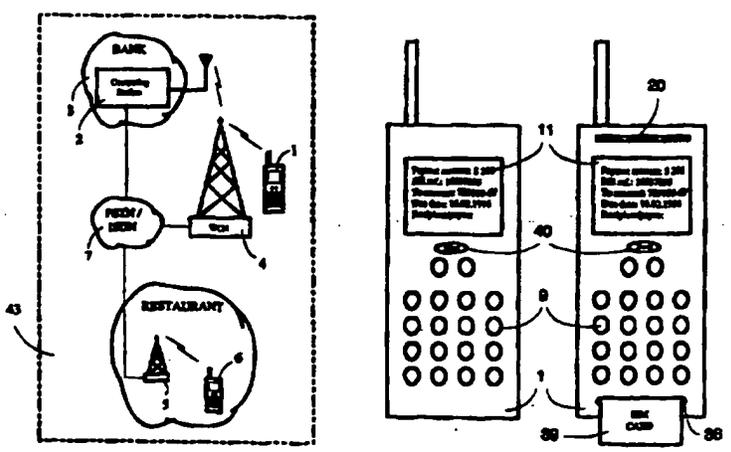




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁶ : G07F 7/08, 19/00, G06F 17/60 // 157:00</p>	<p>A1</p>	<p>(11) International Publication Number: WO 96/13814 (43) International Publication Date: 9 May 1996 (09.05.96)</p>
<p>(21) International Application Number: PCT/FI95/00591 (22) International Filing Date: 25 October 1995 (25.10.95) (30) Priority Data: 945075 28 October 1994 (28.10.94) FI (71)(72) Applicant and Inventor: VAZVAN, Behruz [FI/FI]; Jämärantaival 11 B 53, FIN-02150 Espoo (FI).</p>	<p>(81) Designated States: FI, JP, US, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i> <i>With amended claims and statement.</i></p>	

(54) Title: REAL TIME TELE-PAYMENT SYSTEM



(57) Abstract

This invention is a real time mobile tele-payment system that relates to payments of bills of mobile users, or providing the mobile users with the information about their bank account, the statement of account, or the movement on the account in a real time basis, by using their portable telephones under any wireless telecommunications systems. Certain features of this invention are intended as an expansion of value-added services of currently existing mobile communications systems. This invention also provides the retail and trading businessmen with the possibility to charge their customers, via wireless communications networks and in a real time basis, by using their mobile telephones. In this invention, in order to pay his/her bills, a mobile telephone subscriber enters the payment (bill) information and the payee's account number into the mobile payment part (10) which is included in his/her mobile telephone (1) or (6). After having dialled the telephone number of computing station (2) which is based in the bank (3), the payment information will be sent to the computing station (2) via a mobile communications network (4). In the computing station (2) the calling party's identity will be checked and then the payment will be transferred from the calling party's bank account to the payee's account and then both the calling party and the payee will be informed about the relevant payment. In this invention, the portable telephone is also equipped with a small charge slip printer which can print a receipt for customers of retail businesses.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				

Real Time Tele-payment System

This invention is a mobile payment system that relates to payments of bills of the mobile users, or providing the mobile users with the information about their bank account balance, the statement of account, or the movement on the account in a real time basis, by using their portable telephones under any wireless telecommunications systems.

BACKGROUND OF THE INVENTION

There are several mechanical and electrical payment systems for retail business operations like, for example, what is introduced by US patent US-A-5 202 825, in which a hand-held data terminal generates a record of purchases made by a customer for charging a customer in accordance with customer-indicated payment preferences. In these systems the waiter sends by use of a portable data terminal the customer's order to a customer service station which is a typical cash register based in the restaurant. These systems reduces the time requirements for processing customers at check-out counters in comparison with those of more traditional check-out procedures of the recent past. These systems are only for sending the customer order to the cash register in the retail business.

On the other hand in the fixed telecommunications networks a user (subscriber) can be connected from his personal computer to his/her bank via telephone lines and thereby pay his/her bills. In such systems user must use a data modem between his/her computer and the telephone wire. Another disadvantage of such systems is that in order to pay his/her bills, user must have access to a personal computer connected to the fixed telephone infrastructure, therefore user mobility in such systems is completely limited. Before this invention, there was no solution that provides the portable/mobile telephone users with the possibilities to pay their bills by using their personal portable telephones. There was also no payment system, based on use of portable or mobile telephones, that could provide the retail or trading businesses with the possibility to charge their customers in a real time basis; transferring the charges from the customer's account to the account of the retail businessman. There continues to exist a need to further improve the efficiencies of payment systems.

DESCRIPTION OF THE INVENTION

In order to serve such current need, the present invention provides a new and unique mobile payment system. In the inventive system a portable telephone can be used in order to pay bills or transfer money from a bank account to other, or request the bank

for account information. Certain features of the invention are intended as an expansion of value-added services of currently existing mobile communications systems. This invention addresses needs created by users mobility. For example, suppose that you are travelling and you want to pay a certain bill or transfer some amount of money from your bank account to another person's account but you do not have time for going to the bank or the bank may be closed and you may neither have access to your personal computer (which can be connected to the bank via telephone wire). This invention provides you the possibility to pay your bills, by using your portable telephone while you are in move, regardless of are banks closed or not, regardless of if it is night or weekend etc. This invention also provides the retail businesses (for example restaurants) the possibility to charge their customers, via wireless telecommunications networks, by using only the portable telephones. For example, a waiter in a restaurant, after having entered the amount of payment and customer's information (like account number etc.) to his/her portable terminal can send the payment information to the inventive computing station, which is located in the bank. In the computing station the customer's bank account will be charged in accordance with the payment amount received from the waiter's portable telephone. The most important advantage gained by the inventive system is that all mobile telephone subscribers can pay their bills by using only their normal mobile telephones (in which the mobile payment part is included) and their subscriber identity or codes, without requiring any additional data modem, personal computer, and credit cards etc. In this invention the subscriber identity and codes function as the credit card or bank card of the portable terminal's user.

By implementing the inventive mobile payment system a mobile user (subscriber) can pay all his/her bills and handle all his/her banking issues by only using his/her mobile telephone and subscriber identity or codes, where ever under the coverage of a wireless communications network. These and other improvements and advantages are realised by providing a portable telephone (hereafter called portable terminal) including the inventive mobile payment part, and a computing station which is based in the bank. The present invention will now be described by way of examples with reference to the accompanying drawings, in which:

Fig. 1 is a schematic representation of the inventive Real Time Tele-payment System.

Fig. 2 represents, as an general example, a payment flow diagram between the portable terminal and the computing station, which is located in the bank.

Fig. 3 represents, as an general example, a payment flow diagram in which a mobile user pays his/her bills or request the statement of his/her account by using his/her own

portable telephone. In this figure also the payee is informed about the reception of a payment.

Fig. 4 is a schematic representation of two type of portable terminal: one is a normal portable telephone that includes the inventive mobile payment part, and the other is a portable telephone that includes the inventive mobile payment part, a charge slip printer and a user-friendly SIM card reader (SIM: Subscriber Identity Module).

When a mobile user wants to pay a bill or transfer money from an account to other, he/she enters all information required for payment (like his/her account number, the payee's account number, payment's due date, bill's reference number, etc.) to the mobile payment part of his/her portable terminal 1 (for example through the keypad). As it is the object of this invention, the user's own account information dose not need to be entered into the mobile payment part if the computing station 2, based in the bank 3, can identify the calling party. This needs that the user information (identity) should be confirmed by his/her telephone operator or service provider in a wireless communications network 4 and then be sent to the bank as a confirmation of user (subscriber) identification. More precisely, user identity can be sent by user's telephone operator or service provider to the computing station 2 when portable terminal 1 set-ups a call or a short message to the computing station 2. Monitoring a calling party's subscriber number or information at a receiving terminal is a feature provided by today's digital telephone systems. In this invention, in order to implement such procedure, for example the switching systems at the mobile network side can be used so that only when a user set-ups a call or sends a message (by using short message services of the mobile communications systems) to the computing station 2 his/her identity can be monitored in the computing station 2 in order to identify who is the calling party. Therefore, in this invention the computing station 2 receives at least the confirmed user identity from the user's telephone operator or service provider of a wireless communications network (WCN) 4 in order to identify who is in charge for payment of bills sent by portable terminal 1. Other required information like passwords or access codes to the user's bank account will be sent by user through his/her portable terminal 1. In today's mobile telecommunications systems the user identity, included in his/her SIM card, is checked and confirmed by network 4 every time his/her portable terminal 1 is turned on and attached to the telephone network 4. Since the user identity, transmitted from the portable terminal 1 to the network 4, is completely encrypted and secured therefore the payment messages between portable terminal 1 and computing station 2 are also quite secured because of: first, the security algorithms used in the today's digital wireless telecommunications systems and mobile telephones, and secondly, because of the user's password or access codes used for payment messages in the inventive mobile payment system. All kind of wireless

communications networks can be used in order to communicate the payment messages between the portable terminal and computing station. For example if in the restaurants there is a cordless network like DECT (Digital European Cordless Telephony) 5 then the portable terminal 6 can be connected through such network and PSTN (Public Switched Telephone Network) or ISDN (Integrated Services Digital Network) 7 to the computing station 2.

The payment question-answering procedure between the user and portable terminal 8 is entered by using the user interface 9 and received and handled by the inventive mobile payment part 10. The payment information entering procedure 11 is an interactive procedure between the mobile payment part 10 and the user through user interface 9. Then, the computing station's telephone number will be dialled 12 (either automatically or by user) which after the portable terminal 8 sends the required information for call set-up to the wireless communications network 15 and then payment messages 13 to the computing station 14 via the same network 15. If the portable terminal 8 does not send the user (telephone subscriber) identity to the computing station 14, then the wireless communications network 15 confirms and sends the user identity to the computing station 14 either directly or through the fixed public network 16. The computing station 14 checks the calling party's account and account number of payee (the account to which the payment should be transferred) and then transfers the required amount of payment from the payer's account to the account of payee 17. After that the payment has been completed the computing station 14 sends a message 18 to the portable terminal 8 indicating "payment completed" or if there is not enough credit (money) in the payer's account a "No effects" message 19 will be sent to the portable terminal 8, meaning that the payment can not be accepted. For retail businesses, portable terminal includes also a charge slip printer 20. If the portable terminal receives a "payment completed" command 18, the charge slip printer 20 prints a receipt for the customer. In this invention for the retail and trading businesses, the customer's SIM card 39 is entered in the SIM card reader 36 of the portable terminal 1 (of a waiter in a restaurant, for example) temporary in order to pay the bill. Then the portable terminal 8 will be connected to the wireless communications network 15. The account number of payee (for example account number of the restaurant) can be saved in the memory of his/her portable terminal in order to reduce the information entering procedure of the mobile payment part. This means that only the payment amount should be entered to the mobile payment part. After that the payment amount has been entered to the mobile payment part 10 and the computing station's 14 telephone number has been dialled 12, the wireless network 15 sends the customer's identity, which can be the subscriber identity or a different code.

to the computing station 14. The computing station 14 can identify the calling party (the payer) because it has received the calling party's identity from the wireless network 15 and compared with the calling party's identity based in the computing station 14. Therefore the calling party will be charged for the payment amount received from the portable terminal 8. The subscriber identity sent from the wireless network 15 to the computing station 14 can be different than the payer's identity sent by the portable terminal 8 to the wireless network 15 but both of these identities belong to one user (subscriber). Alternatively the payer's identity, included in his/her SIM card 39 or entered to the portable terminal by using user interface 9, can be sent directly from the portable terminal 8 to the computing station 14. It should be understood that for the simplicity of the description, messages for outgoing call set-up and incoming call or short message services procedures are not explained with details since these procedures are already well known in the mobile communications systems.

Following is an example, in which a mobile user pays his/her bills or transfers money from his/her bank account to other, or ask the bank for statement of account, by using his/her own portable telephone.

First, the payer enters the bill's information 22 (for example: account number of payee, the amount of money which should be transferred, due date of the bill, reference number 11) to the mobile payment part 21 of his/her portable terminal 41. Then, after activating an OK function by user, the mobile payment part dials 23 the telephone number of the computing station located in the bank 24, which after the mobile payment part 21 sends the payment information 25 to the computing station 24, via a wireless communications network (WCN) 26 and fixed network 27 (PSTN/ISDN). Then, computing station 24 transfers the amount of payment, mentioned on the bill, from the payer's account to the payee's account 28. Then, computing station 24 sends a "Payment Completed" message 29 to the portable terminal's mobile payment part 21. If the payee has also a portable terminal 37, then also his/her mobile payment part 42 would receive a "Payment Reception message" 30, from computing station 24, indicating the amount of payment, the payer and the payment date. However, before dialling the number of computing station, the mobile payment part may ask the payer (the user of portable terminal) "Any other payment ?" 31. The answer can be respond by activating "Yes/No" function 32 or OK function of the mobile payment part 21. Then the user can enter another bill information to the mobile payment part 21 and when all information required by mobile payment part has been provided, the telephone number of computing station 24 will be dialled 23. After this, all bills information (payment messages) will be sent to the computing

station in the bank 24 as explained above. Furthermore, there is a command 33 "Send the Statement of Account" in the mobile payment part 21 for requesting the account balance, the statement of account, or the movement on the account from the computing station 24. When a user selects such command 33, the mobile payment part 21 sends this message 33, either by setting up a call or by using the short message facilities of mobile communications networks 26 to the computing station 24. Then computing station 24 sends the required account balance or the statement of account 34 to the mobile payment part 21 of the portable terminal 41. The computing station 24 also sends a "Monthly Statement of Account" 35, to the portable terminals 41, 42 once or twice per month. Then portable terminal's printer 38 can print it for the user to be filed as a record, if required.

Following is an example in which the payee (for example a restaurant or a retail seller) has a portable terminal by which the payer's (a customer) account can be charged.

Suppose that a customer wants to pay his/her bill in a restaurant for the service he/she has received. The customer can give his/her SIM card 39 or credit card to the waiter to be entered to the waiter's portable telephone 1, 8. Then waiter dials the telephone number of computing station 14, or the number will be dialled automatically after the SIM card 39 or credit card has been read by the SIM card or credit card reader 36 of the waiter's portable terminal. For example the telephone number of computing station 14 can be saved in the memory of the portable terminal of waiter, and every time a customer's SIM or credit card 39 is entered to the portable terminal 1, the portable terminal automatically contact the computing station 14, after having registered in the network 15. In the bank, the computing station 14 checks the account information of payer (a customer) and then transfers the transaction amount (the sum on the bill) to the payee's (the restaurant) account 17. If the payer's account do not have enough credit (money) the portable terminal 8 may receive a "No effects" message 19, or the bank may pay the transaction's amount on behalf of the payer and then later charge the payer or his/her bank for the prepaid transaction. On the other hand if the payer's account information (account number, account identity) is false the computing station 14 may send a "transfer not accepted" message to the payee's portable terminal, which means that the payer (customer) should pay the amount of transaction in cash. If the portable terminal receives from the computing station 14 a "payment completed" message 18, then the charge slip printer 20 prints a receipt for the customer, as explained in the first example.

It should be considered that in all above-mentioned examples, payment messages can be sent and received either by setting up a call between the portable terminal and computing station or by using short message services facilities of the wireless communications networks.

In the current mobile communications systems, like GSM, there is a facility called "Short Message Services, (SMS)". In SMS a mobile telephone user can send short messages to another subscriber without setting up an interactive call. In order to send the payment messages by SMS, the software of SMS installed in the portable terminal can be modified so that it can also handle the payment parameters and/or commands of the inventive mobile payment part 10. Then by using the SMS services of the wireless communications network 15, the bill's information 13 can be sent to the computing station 14. When computing station 14 receives such payment message 13 sent by SMS, it also generates a message to be sent to the portable terminal in order to inform it if the payment has been completed 18 or not 19. However, if a user wants to pay many payments (bills) at once and receive also balance or statement of his/her bank account from the computing stations, such long message can be divided to smaller parts and then be combined at the portable terminal or computing station. This means that each bill information can be sent separately using the short message services. This action is transparent to the user of portable terminal. For example several payment information can be entered to the mobile payment part 10. Then when user selects the "Send" function 40 on the portable terminal 1, each bill will be sent by one short message in accordance of short messages length. For example, a short message may not include more than 100 letters. If a payment message or the statement of account (sent by computing station) needs more than the assumed 100 letters, then such long information will be divided into two or several short messages and then will be sent one by one to the portable terminal or computing station.

In this invention computing station can send and receive messages either via PSTN (Public Switched Telephone Network) and ISDN (Integrated Services Digital Network) and other fixed networks or via only a wireless communications network. The computing station includes all means for transmitting and receiving payment and banking messages via the wireless networks.

It is to be understood that various changes and modifications can be made to alter the specifically described structure or methods of operation of the preferred embodiment without departing from the spirit and scope of the invention. This invention is to be defined only by the scope of the claims appended hereto.

Claims

1. A mobile payment system (43), characterised in that it is comprised of:

- at least one portable terminal (1, 6, 8), such terminal including a mobile payment part (10, 21) and other means for entering, transmitting, receiving and printing of information relating to: the payments of bills of the telephone subscriber or the user of said portable terminal; transferring of money from the bank account of the subscriber or user to the others account; sending and receiving payment messages (13, 18, 19, 25, 29) or messages including the account balance, the statement of account, or the movement on the bank account (33, 34, 35) of the telephone subscriber or the user of the portable terminal (41, 37);

- at least one computing station (2, 14, 24) which is located in the bank (3), said computing station including means for communicating with said portable terminal and for transferring the amount of payment (money) from the bank account of portable terminal's user and/or telephone subscriber to another bank account (17, 28), or from a customer's bank account, whose account information is entered into said portable terminal, to the calling party's account; and to receive and send messages about the account balance, the statement of account, or the movement on the bank account (33, 34, 35) of the portable terminal's subscriber or user;

- at least one wireless communications network (4, 15, 26) through which said portable terminal can send and receive to or from said computing station said payment messages or messages about the account balance, the statement of account, or the movement on the bank account of said portable terminal's subscriber or user.

2. A mobile payment system (43) according to claim 1, characterised in that said at least one portable terminal (1, 6) is a first plurality of portable terminals, and in which the number of said portable terminals in said first plurality of portable terminals is greater than said at least one computing station (2).

3. A mobile payment system (43) according to claim 1 and 2, characterised in that the payments or bills of a mobile telephone subscriber can be paid by entering the subscriber identity and codes into said portable terminal (1, 6, 8, 41) and the bill's information, including the payee's bank account number, the amount of payment, bill's due date and reference number into the mobile payment part (10, 21) of said portable terminal, and by setting up a call or a short message to the bank's computing station

(2, 14, 24) and sending the payment (bill's) messages (13, 25) to said computing station (2, 14, 24).

4. A mobile payment system (43) according to claim 1, 2 and 3, characterised in that said at least one portable terminal (1, 6) comprises all means for transmitting and receiving payment messages to or from said computing station (2); and that:

- said portable terminal includes a mobile payment part (10, 21) for handling the payment information (11, 22, 31, 32) entered by user to said portable terminal, and that said payment information can be saved into the memory of said portable terminal and be sent to said computing station, whenever required; and that:

- said portable terminal receives a message (18, 19, 29) from said computing station indicating that either the payment or transferring of the required amount of payment from the payer's to the payee's bank account has been accepted and/or completed or not.

5. A mobile payment system (43) according to claim 1, 2, 3 and 4 characterised in that the user of said portable terminal can enter more than one payment or bill information to the mobile payment part (10, 21) , and that after that telephone number of said computing station based in the bank (2, 14, 24) has been dialled (12, 23) either manually or automatically, all required payment information (13, 25) will be sent to said computing station; and that:

- said portable terminal can send payment (bill's) information (13, 25), handled in mobile payment part (10, 12), to the computing station (14, 24) and receive the required payment messages (18, 19, 29) from said computing station by setting up a call or using the Short Message Services (SMS) of the wireless communications network (4, 15, 26); and that

- said portable terminal's subscriber information can be sent from the user's telephone operating network (4, 15, 26) to the computing station (2, 14, 24); and that

- said portable terminal includes a charge slip printer (20, 38) that can print all payment information and the information received from said computing station for user of said portable terminal, and that,

- said mobile payment part (10, 21) can be included into any kind of digital or analogue portable telephone that is capable of operating in cellular communications systems.

6. A mobile payment system (43) according to claim 1 - 5, characterised in that said computing station (2, 14, 24) after receiving a payment message (13, 25) from said portable terminal (1, 6, 8, 41), checks and charges the payer's account (17, 28) in accordance with the payment amount received from said portable terminal and then sends a message (18, 19, 29) to said portable terminal (8, 41, 37) in order to indicate that payment has been accepted and/or completed or indicating that there is not enough credit in the payer's account; and that:

- said computing station (2, 14, 24) can receive or send payment messages (18, 19, 29, 30) or other banking messages (33, 34, 35) to said portable terminal (1, 6, 8, 41) via either fixed and wireless communications networks (4, 7, 15, 16, 26, 27) or via only wireless communications network (15, 26); and that,

- said computing station (2, 14, 24) can receive the payer's information and identity either from the payer's telephone operator or service provider through wireless communications network (4, 15, 26) when payer telephones or send messages (13, 25) to said computing station (2, 14, 24) or from the payer's portable terminal (1, 6, 8, 41); and that, the payer's information received from said payer's telephone operator or service provider or from said portable terminal may include payer's subscriber information or identity or any other required information; and that,

- said computing station can monitor the subscriber information or other identity, received from said payer's telephone operator or service provider or portable terminal, and based on said subscriber information or other identity and account number transfer the required amount of payment (money) from the payer's account to any other required account; and that,

- said subscriber information or identity will be confirmed by subscriber's telephone operator or service provider (4, 15, 26) and said confirmed information will be sent to said computing station (2, 14, 24) in which the subscriber identity will be checked (17, 28) and based on that, the received payment message (13, 25) can be accepted and a payment completed message (18, 29) will be sent to said portable terminal (1, 6, 8, 41); and that,

- said computing station (2, 14, 24) can send or receive payment messages (13, 18, 19, 25, 29, 30, 33, 34, 35) to or from the portable terminals (1, 6, 8, 41, 37) of both the payer and the payee; and that,

- said computing station (2, 14, 24) is equipped with all means for transmitting and receiving messages via any wireless communications network, to or from said portable terminal (1, 6, 8, 41, 37).

7. A mobile payment system (43) according to claim 1 - 6, characterised in that the mobile payment part (10, 21) may ask the user to enter all payment information

(11) such as payee's account number, bill's reference number, bill's due date, the amount of payment and other required information; and that:

- said mobile payment part (10, 21), after receiving all information about a payment or a bill from the user through user interface (9), may ask the user of said portable terminal "any other payment ?" (13) indicating dose user wants to pay another bill or payment; and that,

8. A mobile payment system (43) according to claim 1 - 7, characterised in that said portable terminal (1, 6, 8, 41) can be used in order to pay the bills of any mobile telephone subscriber by entering each subscriber's identities and codes into said portable terminal either by using the portable terminal's user interface (9) or the SIM card (39) and card reader (36); and that:

- said mobile telephone subscriber's codes can be different than said subscriber's identities; and that said subscriber codes can be included both in the subscriber's SIM card (39) and said computing station (2) located in the bank (3); and that:

- said portable terminal (1, 6) can be used in order to charge customers, in retail or trading businesses, by entering the customers' telephone SIM card (39) into said portable terminal (1, 6) and by using the telephone subscriber identities of each customer as an identification for payment; and that:

- after that said customer's SIM card (39) has been entered to said portable terminal (1, 6, 8), said portable terminal will be re-connected to the wireless communications network (4, 15) in order to check the subscriber identity, which after the customer's (subscriber's) bank account can be charged by sending payment messages (13) to the computing station (2, 14).

AMENDED CLAIMS

[received by the International Bureau on 25 March 1996 (25.03.96);
original claims 1 and 3-8 amended; new claims 9 and 10 added;
remaining claims unchanged (8 pages)]

1. A mobile payment system (43), utilizing the Short Message Services (SMS) facilities of mobile communication networks such as GSM (Global System for Mobile Communications), and subscriber identity such as SIM card (Subscriber Identity Module), and a new mobile-telephone-based functionality and mobile network architecture characterized in that it is comprised of:

- at least one portable terminal (1, 6, 8), such terminal utilizing the inventive Mobile Payment Part (10, 21), which provides a function and SMS-based adaptation and application part integrated into said portable terminal to provide at least an alphanumeric payment (bill) inquiry (e.g. 11), and including other means for entering, transmitting and receiving, and printing of the information mainly related to: the payments of bills of the telephone subscriber (1, 6, 8); transferring of money from the bank account of the subscriber or user to the others account; sending and receiving payment messages (13, 18, 19, 25, 29) or messages including the account balance, the statement of account, or the movement on the bank account (33, 34, 35) etc. of the telephone subscriber of the portable terminal (41, 37) without requiring to use any additional data modem to be used in conjunction with said portable terminal for transmission and reception of said payment etc. messages;

- at least one computing station (2, 14, 24) which is located in the bank (3), as it is the object of the architecture of the inventive payment system (43), said computing station includes all information about the portable telephone subscriber data which is connected to the subscriber's bank account in the same bank wherein computing station is located, and said computing station includes means for communicating with said portable terminal (4) and transferring the amount of payment (money) from the bank account of the payer (i.e. the calling subscriber) to another bank account (17, 28), and to receive and send messages about the payments, account balance, the statement of account, the movement on the bank account (33, 34, 35) or other banking messages etc. of the calling subscriber via SMS facilities of the wireless communication network (4):

- at least one wireless communication network (4, 15, 26) equipped with Short Message Services (SMS) infrastructure through which said portable terminal (1, 6, 8) can send and receive to or from said computing station said payment messages or other banking messages etc. and that said wireless communication network can confirm (i.e. authenticate) the subscriber identification for said computing station, whenever required or transfer the subscriber data received from said portable terminal directly to said computing station, in which the subscriber data can be compared with the subscriber data already recorded there.

2. A mobile payment system (43) according to claim 1, **characterized** in that said portable terminal (1, 6) is a first plurality of portable terminals, and in which the number of said portable terminals in said first plurality of portable terminals is greater than said at least one computing station (2).

3. A mobile payment system (43) according to claim 1, 2, **characterized** in that said portable terminal (1, 6, 41) comprises all means for transmitting and receiving payment etc. messages to or from said computing station (2) or other portable terminal (37); and that,

- said portable terminal includes the inventive Mobile Payment Part (10, 21) which is a short-message-based adaptation and application part for handling, dividing or connecting the payment etc. information (11, 22, 31, 32), and that said payment etc. information can be saved into the memory of said portable terminal and be sent to said computing station, whenever required; and that,

- after that portable terminal has been registered into the mobile network (4), the payments or bills of the mobile telephone subscriber (1, 6) can be paid by entering the bill's information such as the payee's bank account number, the amount of payment, bill's due date and reference number etc. into the Mobile Payment Part (10, 21), and by sending the short messages (e.g. 13, 33) to the bank's computing station (2, 14, 24) via SMS facilities of the mobile network (4) and receiving messages such as (18, 19, 30, 34, 35 etc.).

- said portable terminal receives a message (e.g. 18, 19, 29) from said computing station indicating that either the payment or transferring of the required amount of money from the payer's to the payee's bank account has been accepted and/or completed or not; and that,

- said portable terminal includes a charge slip printer (20, 38) that can print all payment information and the information received from said computing station for user of said portable terminal, whenever required.

4. A mobile payment system (43) according to claims 1, 2, 3, **characterized** in that the computing station (2, 14, 24) after receiving a payment message (13, 25) from said portable terminal (1, 6, 8, 41), checks and charges the subscriber's (payer's) account (17, 28) in accordance with the payment amount received from said portable terminal and then sends back a message (e.g. 18, 19, 29) including all information about the payment (e.g. bill reference, payer, amount etc.) to said portable terminal (8, 41, 37) in order to indicate that the payment has been accepted and/or completed or indicating that there is not enough credit in the payer's account; and that:

- said computing station (2, 14, 24) can receive or send payment messages (e.g. 18, 19, 29, 30) or other banking messages (e.g. 33, 34, 35) or any other message to said portable terminal (1, 6, 8, 41) via SMS of a mobile communication network (4, 5) through either fixed and wireless communication networks (4, 5, 7, 15, 16, 26, 27) or via only wireless communications network (4, 15, 26); and that.

- said computing station (2, 14, 24) can receive the payer's identity either from the payer's telephone operator system (4, 15, 26) when payer sends messages (e.g. 13, 25) to said computing station (2, 14, 24) or from the payer's portable terminal (1, 6, 8, 41); and that said payer's data received from said payer's telephone operator or from said portable terminal may include the payer's subscriber data or identity parameters or any other required information; and that.

- said subscriber data can be confirmed (i.e. authenticated) and secured either in the databases and infrastructure of the subscriber's telephone operator (4), or in said computing station, for example, by utilizing the algorithms used in mobile communication systems such as those of the GSM; and that.

- after that the subscriber data, communicated between said portable terminal and wireless communication network or directly between said portable terminal and computing station has been authenticated, the Mobile Payment Part (10) of the portable terminal or said computing station can send and/or receives payment etc. messages through SMS of a mobile communication network (4); and that.

- said computing station can monitor the subscriber identity, number etc., received alternatively from said payer's telephone operator or said portable terminal, and based on said subscriber identity and/or number and checking of his/her bank account number transfer the required amount of payment (money) from said payer's account to any other required account; and that.

- said subscriber data can alternatively be confirmed or sent by the subscriber's telephone operating network (4, 15, 26) to the computing station (2, 14, 24); as a confirmation of subscriber identification, enabling said computing station to compare the received subscriber data with the data already recorded in said computing station, and when subscriber data is compared and accepted by said computing station, the portable terminal can send payment messages to said computing station; and that.

- said subscriber data may include the subscriber telephone number, confirmed by mobile operator (4), or it may consist of the subscriber identity incorporated in SIM card, or any other code; and that.

- said computing station (2, 14, 24) can send or receive payment messages (e.g. 13, 18, 19, 25, 29, 30, 33, 34, 35) to or from the portable terminals (1, 6, 8, 41, 37) of both the payer and the payee; and that.

- said computing station (2, 14, 24) is equipped with all means for wired or wireless transmission and reception of messages communicated between said computing station (2), wireless communications network (4 or 5), and said portable terminal (1, 6, 8, 41, 37).

- said computing station may send e.g. a monthly report (e.g. 35) to said portable terminal (1, 6, 37, 41) to be displayed or printed (20, 38), for said subscriber, as a receipt and bank report for payments (bills, etc.) charged from the subscriber/payer account to the other subscriber/payee account, by said computing station.

5. A mobile payment system (43) according to any preceding claims, **characterized** in that the subscriber, for example, a waiter etc. in a restaurant can send the payment messages (e.g. a bill) by using the inventive portable terminal (1, 6, 8) either to the computing station (2, 14) or directly to the customer's portable terminal (e.g. a mobile telephone integrated with the inventive Mobile Payment Part), via SMS facilities of mobile communication network (5, 4), which after the payment can be accepted by said customer and be sent to the computing station (2) in which the payment procedure will be completed and then a message (including the bill's information) will be sent to both customer's and waiter's portable terminals indicating that either the payment has been completed and/or accepted (29, 30) or refused (19); and that:

- said waiter etc. or customer can enter the customer's identity code to said waiter's portable terminal, by using user interface (9), and then send the bill together with the customer's code to the computing station, which after said computing station generates a message and sends it to the customer's portable terminal to be accepted by the customer, and that after that the payment has been completed in the computing station, the computing station can send a message such as "Payment Reception" including all information about the payment (e.g. bill reference, payer, payment amount etc.) to the payee's terminal indicating that the payee has received the payment; and that.

- said portable terminal (1, 6) can be used in order to charge customers, in retail or trading businesses, by entering alternatively the customers' telephone SIM card (39) into said portable terminal's SIM card reader (36) and by using the telephone subscriber identity of each customer as an personal identification for payment; and that:

- after that said customer's SIM card (39) has been entered to said portable terminal (1, 6, 8), said portable terminal will be re-registered to the wireless communications

network (4, 15) and/or said computing station in order to check the subscriber identity, which after the customer's (subscriber's) bank account can be charged by sending payment messages (e.g. 13) to the computing station (2, 14).

6. A mobile payment system (43) according to any preceding claims, **characterized** in that whenever the subscriber turns on his/her portable terminal (1, 6) the Mobile Payment Part (10) sends the subscriber data, that can be included in the SIM card, to the computing station (2, 14) through an available wireless communication network (4, 5), and after that registration process between said computing station, said wireless communication network and said portable terminal (1, 6, 8) has been completed said portable terminal can have access to said wireless communication network through which it can send and/or receive payment, banking etc. messages to/from said computing station, and also be able to use the telecommunications services like voice etc. of said wireless communication network; and that.

- after said portable terminal has been registered in said wireless communication network (4) or computing station (2), the subscriber of said portable terminal can send and/or receive banking messages (e.g. 33, 34, 35) or can pay his/her bills by sending and receiving the payment etc. messages (e.g. 13, 18, 19, 25, 29) to the computing station (2, 14) or to another portable terminal, through the Mobile Payment Part (10) of his/her portable terminal; and that.

- said subscriber data can be a data which is recorded only in the SIM card and in said computing station that is located in the bank; and that.

- said subscriber data can be either similar to or different from that subscriber identity which is incorporated in the subscriber's telephone SIM card provided by mobile operators (4); and that.

- said subscriber data can be alternatively sent to said computer station after that registration of said portable terminal into said wireless communication network (4, 5,) has been completed, which after the subscriber can send and/or receive payment/bill messages to said computing station via SMS of said wireless communication network (4, 5,).

7. A portable terminal (1, 6, 8, 41, 37) according to any preceding claims, **characterized** in that it includes the inventive Mobile Payment Part (10, 21) which for each payment procedure may ask the subscriber (i.e. the payer) to enter all payment information (e.g. 11) such as payee's account number, bill's reference number, bill's due date, the

amount of payment and other information included in the bill or required for payment procedure: and that:

- said Mobile Payment Part (10, 21), after receiving all information about a payment or a bill from the user through user interface (9), may ask the subscriber of said portable terminal e.g. "Any other payment ?" (13) indicating dose subscriber wants to pay another bill or payment, and that after this message subscriber can enter other payment information into said Mobile Payment Part: and that,

- said portable terminal (1, 6, 8, 41) can be used in order to pay the bills of any mobile telephone subscriber by entering each subscriber's identities and codes into said portable terminal either by using the portable terminal's user interface (9) or by entering the SIM card (39) and card reader (36): and that:

- more than one payment or bill etc. data can be entered into said Mobile Payment Part (10, 21) of said portable terminal, and that after that telephone number of said computing station based in the bank (2, 14, 24) has been dialed (12, 23) either manually or automatically, all required payment information (e.g. 13, 25) will be sent to said computing station via SMS facilities of the mobile network (4, 5, 15, 26): and that,

- said portable terminal (1, 6, 8, 37, 41), includes all means of a mobile/cellular/cordless telephone for receiving and transmitting voice and data so that said portable terminal can function both as a mobile payment device and as a mobile/cellular/cordless telephone without requiring any data modem to be used in conjunction with the transmission and reception of payment etc. messages: and that,

- said Mobile Payment Part (10, 21) can be integrated into any kind of portable telephone that is capable of operating in cellular communications systems: and that,

8. A portable terminal (1, 6, 8, 41, 37) according to any preceding claims, **characterized** in that a small printing device (20, 38) is integrated into said portable terminal (1, 6, 37, 41) for printing any data received from computing station (2) or other portable terminals or any other source or the messages entered into said Mobile Payment Part (10, 11, 21) by its user or any other short messages received by said portable terminal.

9. A Mobile Payment Part (10) according to any preceding claims, **characterized** in that it is a component and function integrated into the portable terminal (1, 6), said Mobile Payment Part provides a payment (bill) inquiry (11) procedure including for example questions (such as payment amount, Bill reference, Receiver's account number, Due date, Recipient etc.) which can be displayed on the display (6) and which can be answered by the

user of the portable terminal through the user interface (9) and such payment information can be saved into the memory of the portable terminal or be sent to the computing station (2) or another portable terminal via SMS; and that

- said Mobile Payment Part (10) can be either integrated into said portable terminal as a component including the required soft-ware for providing said bill inquiry (e.g. 11), or said Mobile Payment Part can be integrated into the SIM card (i.e. Subscriber Identity Module) to provide said bill inquiry whenever subscriber wants to pay a bill or perform a payment; and that.

- said Mobile Payment Part is a function and SMS-based adaptation, integrated into said portable terminal or alternatively into said SIM card to provide an alphanumeric payment (bill) inquiry (11) procedure; and that.

- said Mobile Payment Part (10) can divide and split any long data of any length, for example e-mails done in a personal computer etc. which can be connected to said portable terminal into several short messages and send them to other portable terminals/telephones (e.g. 1 or 6) or to said computing station (2) via SMS facilities of a wireless communication network (e.g. 4 or 5) without requiring any data modem to be connected between the portable terminal and said personal computer, so that said Mobile Payment Part divides such e-mail to several short messages in a numbering sequence, for example, first short message, second short message etc.; and that.

- said Mobile Payment Part (10) is able to connect several short messages originated from a long data of any length e.g. an e-mail according to said short messages' numbers defined in the sender's portable terminal (e.g. 1) and their sender's identity (e.g. subscriber number), and put them into the original order and configure said original long data, which can be a long information sent by computing station or another portable terminal (e.g. 6) or any other source equipped with the inventive Mobile Payment Part (10) via SMS facilities of a mobile communication system (4), and then display said original data (e.g. the e-mail) on the display of the portable terminal (1, 6) or forward it to a separate monitor or personal computer without requiring any data modem to be used between said portable terminal and said personal computer; and that.

- all short messages which are resulted from a longer data and received by said portable terminal (1 or 6) or computing station (2) may contain a short message number which is unique for each message and is defined according to their dividing sequence; and that.

- all short messages which are divided from a longer data and received by said portable terminal (1 or 6) or computing station (2) , through SMS infrastructure (4 or 5).

may contain both the sender's and receiver's identity number (e.g. payer's and payee's subscriber numbers), which can be added to each short message either at said message sending portable terminal (e.g. 1) or at the wireless communication network's SMS facilities (4); and that.

- all short messages which are divided from a longer data and received by said portable terminal (1 or 6) or computing station (2), through SMS infrastructure (4 or 5), may be connected according to their sender's identity and their arrival time to the SMS facilities of a mobile communication system (4) or their sending time from the portable terminal or computing station or any other source: and that such sending or arrival time can be defined either at said portable terminal (e.g. 1), which sends the messages, or at the SMS facilities of the wireless communication network (4).

10. A mobile payment system (43) according to any preceding claims, **characterized** in that the telephone calls made by portable terminal (1, 6) can be charged simultaneously after each or several calls, from the subscriber's bank account (i.e. payer's account) to the wireless communication operator's (4 or 5) account, so that said operator can send the bills relevant to the telecommunications services used by said subscriber, directly to said computing station (2); and that,

- said computing station can include either the subscribers' data and bank account information or both the subscribers' and said wireless communication operator's data and bank account information so that the subscribers' all telephone calls can be charged directly from the subscriber's account to said wireless communication operator's bank account, by said computing station: and that.

- said computing station may send e.g. a monthly report (e.g. 35) to said portable terminal (1, 6, 37, 41) to be displayed or printed (20, 38), for said subscriber, as a receipt against charged calls or any telecommunications services used by said subscriber and charged from said subscriber bank account to the wireless communication operator's (4 or 5) bank account, by said computing station.

STATEMENT UNDER ARTICLE 19

Hereby we would like to file and publish the attached Amendment together with the above application. The claims filed are amended in order to better define the scope of the claims for the purposes of provisional protection. All claims are amended after that International Searching Report was received by the applicant so that the amended claims define the scope of the claims mainly based on using the second alternative (i.e. Short Message Services facilities, see page 7 of description). Moreover, it was noticed that the filed claims could not cover all objects of the above-mentioned application without applying for amendment. All claims amended here fall into the description of the invention, and go not beyond the disclosure in the above international application as filed. The differences between the claims as filed and as amended are indicated in the next page.

FIG 2

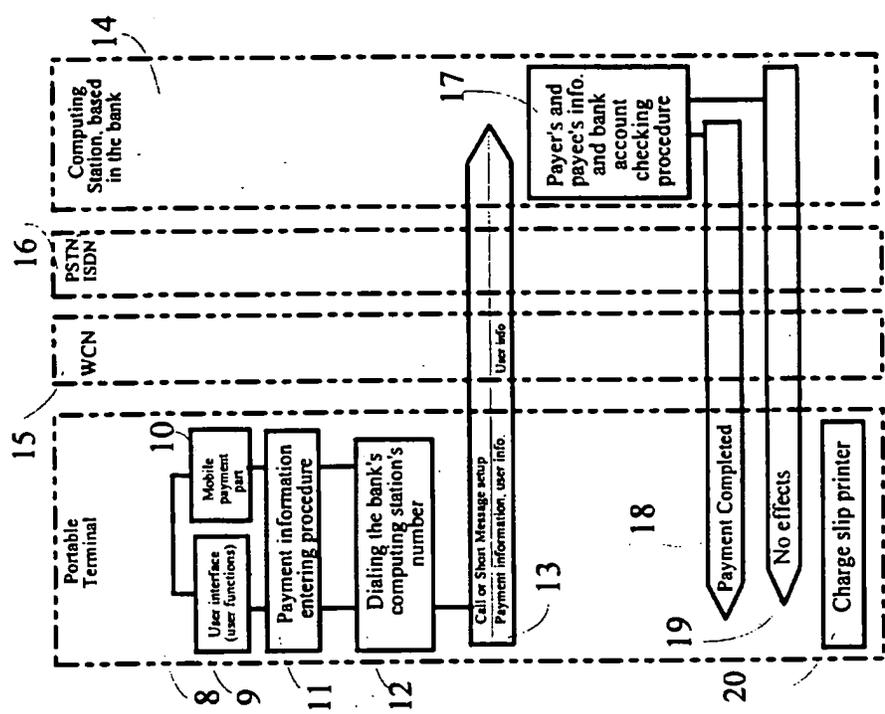


FIG 1

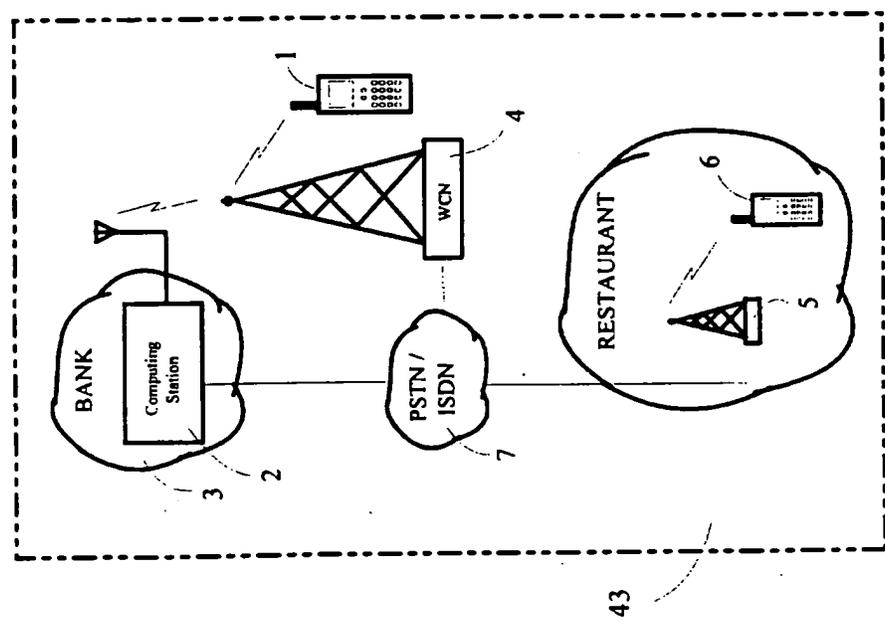


FIG 4

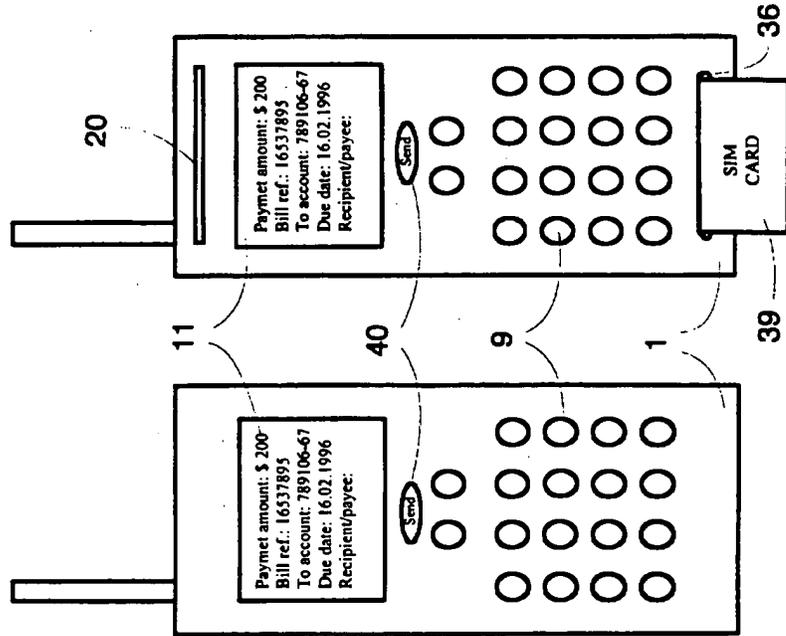
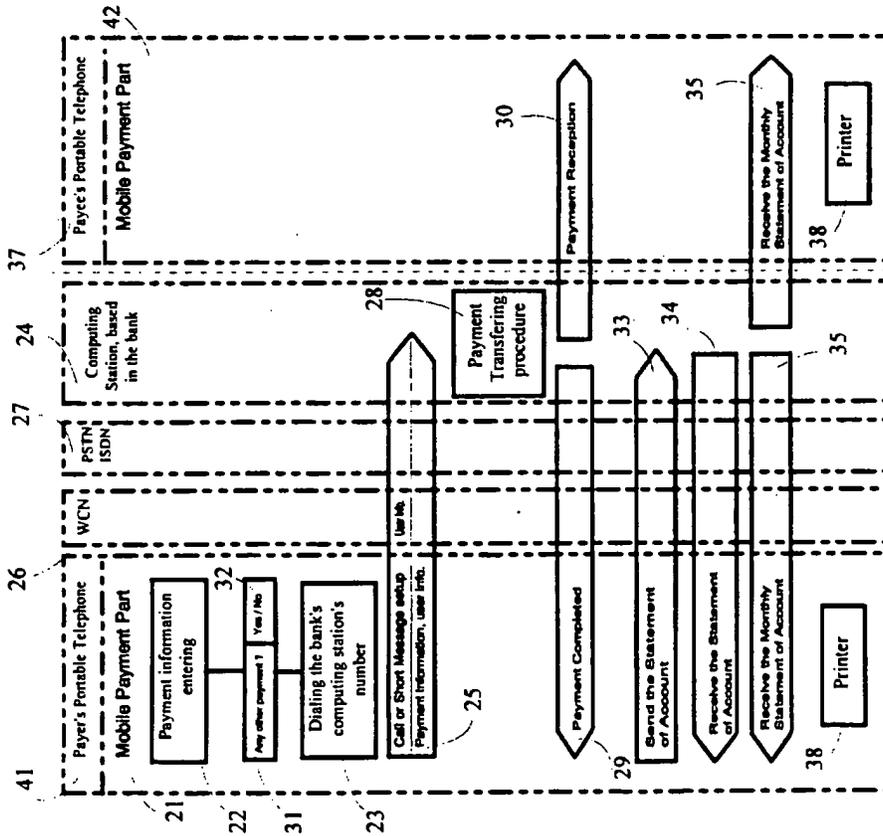


FIG 3



INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI 95/00591

A. CLASSIFICATION OF SUBJECT MATTER		
IPC6: G07F 7/08, G07F 19/00, G06F 17/60 // G06F 157:00 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
IPC6: G07F, H04M, H04Q		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
SE,DK,FI,NO classes as above		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 9411849 A1 (VATANEN, H.T.), 26 May 1994 (26.05.94) ----- -----	1-8
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
<p>* Special categories of cited documents:</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>		
Date of the actual completion of the international search		Date of mailing of the international search report
1 March 1996		04-03-1996
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86		Authorized officer Jan Silfverling Telephone No. +46 8 782 25 00

INTERNATIONAL SEARCH REPORT
Information on patent family members

05/02/96

International application No.
PCT/FI 95/00591

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO-A1- 9411849	26/05/94	NONE	



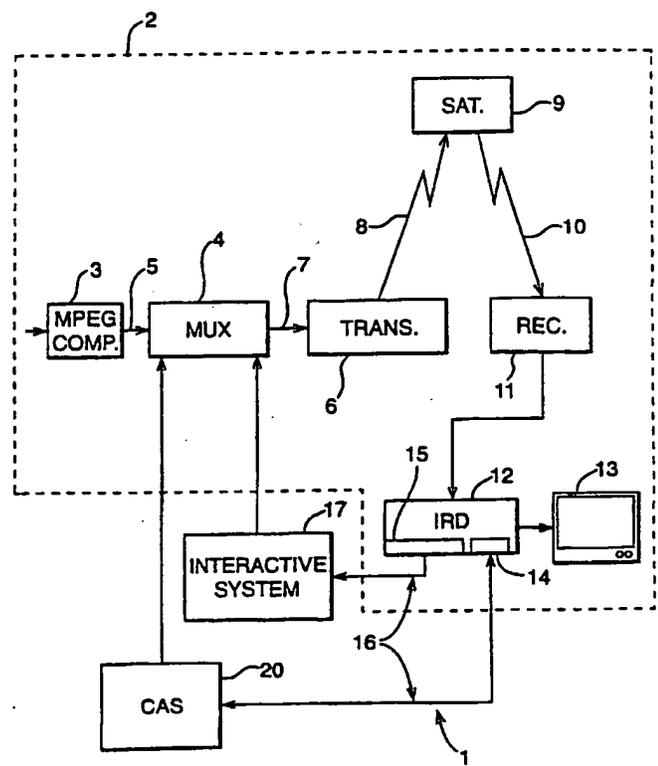
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁷ : H04N 7/16, 7/167</p>	<p>A1</p>	<p>(11) International Publication Number: WO 00/46994 (43) International Publication Date: 10 August 2000 (10.08.00)</p>
<p>(21) International Application Number: PCT/IB00/00163 (22) International Filing Date: 4 February 2000 (04.02.00) (30) Priority Data: 99400261.6 4 February 1999 (04.02.99) EP (71) Applicant (for all designated States except US): CANAL+ SOCIETE ANONYME [FR/FR]; 85/89, quai André Citroën, F-75711 Paris (FR). (72) Inventor; and (75) Inventor/Applicant (for US only): MAILLARD, Michel [FR/FR]; 42, avenue du Maréchal Leclerc, F-28130 Maintenon (FR). (74) Agents: COZENS, Paul, Dennis et al.; Mathys & Squire, 100 Gray's Inn Road, London WC1X 8AL (GB).</p>	<p>(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p>Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>	

(54) Title: METHOD AND APPARATUS FOR ENCRYPTED TRANSMISSION

(57) Abstract

A method and apparatus for encryption of data between a first device (12) and a second device (30), in which one or more precalculated key pairs (41) are stored in a memory of the first device (12), the or each key pair comprising a session key and an encrypted version of the session key. The encrypted version is passed to the second device (30), which decrypts (42) the session key, this session key being thereafter used to encrypt data communicated from the second device (30) to the first device (12) and/or vice versa. The invention is particularly applicable to a digital television system in which data, notably control word data, is to be communicated in encrypted form between a decoder and an associated portable security module.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

METHOD AND APPARATUS FOR ENCRYPTED TRANSMISSION

The present invention relates to a method and apparatus for encryption of messages between two devices, for example a decoder and a portable security module in a digital television system.

Transmission of encrypted data is well-known in the field of pay TV systems, where scrambled audiovisual information is usually broadcast by satellite to a number of subscribers, each subscriber possessing a decoder capable of descrambling the transmitted program for subsequent viewing.

In a typical system, scrambled data is transmitted together with a control word for descrambling of the data, the control word itself being encrypted by a so-called exploitation key and transmitted in encrypted form. The scrambled data and encrypted control word are then received by a decoder having access to an equivalent of the exploitation key stored on a portable security module such as a smart card inserted in the decoder. The encrypted control word is then decrypted on the smart card and subsequently communicated to the decoder for use in descrambling the transmitted data.

In order to try to improve the security of the system, the control word is usually changed every ten seconds or so. This avoids the situation with a static or slowly changing control word where the control word may become publicly known. In such circumstances, it would be relatively simple for a fraudulent user to feed the known control word to the descrambling unit on his decoder to descramble the transmission.

Notwithstanding this security measure, a problem has arisen in recent years where the stream of control words sent during a broadcast becomes known through monitoring of data communicated at the interface between the smart card and decoder. This information may be used by any unauthorised user who has recorded the still-scrambled broadcast on a video recorder. If the film is replayed at the same time as the stream of control words is fed to the decoder, visualisation of the broadcast

becomes possible. This problem has further been exacerbated with the rise of the internet and it is now common to find any number of internet sites that list the stream of control words emitted during a given transmission.

5 The European patent application PCT WO 97/3530 in the name of Digco addresses this problem by proposing a solution in which the control word stream passed across the interface between the smart card and decoder is itself encrypted with a session key. The session key is generated randomly by the decoder and encrypted with a second key held in the decoder and corresponding to a public key used with a private/public
10 encryption algorithm. The associated smart card possesses the necessary private key to decrypt the session key, which is thereafter used by the smart card to encrypt the control word stream sent from the smart card to the decoder.

As will be appreciated, the use of a locally generated session key to encrypt the
15 control word stream means that the encrypted stream cannot thereafter be fed into another decoder for use in descrambling the data since each decoder will possess a different session key for use in decrypting the control word stream sent from the smart card.

20 Whilst this solution provides a higher level of security than conventional systems there are nevertheless a number of disadvantages associated with this system.

Notably, the use of a public/private key algorithm is effectively obligatory in such a system since it is not desirable for security reasons to store both a symmetric key and
25 the associated algorithm in the decoder, due to the ease in which this information may be extracted from a decoder memory. This problem does not arise in the case of a public key, since possession of this key does not enable decryption of private key encrypted messages.

30 It is one object of the present invention to provide a more adaptable alternative to the above known system. However, the invention is not limited to the field of decoder security and, as will be described below, may be applied to a number of other

situations in which secure communication of data is required.

A first aspect of the present invention provides a method of encryption of data communicated between a first and second device, wherein at least one precalculated
5 key pair is stored in a memory of the first device, said at least one key pair comprising a session key and an encrypted version of the session key prepared using a transport key, the encrypted version of the session key being subsequently communicated to the second device which decrypts the encrypted version using an equivalent transport key stored in its memory such that data communicated from at
10 least the second to the first device may thereafter be encrypted and decrypted by the session key in the respective devices.

A preferred embodiment provides a method of encryption of data communicated between a first and second device, characterised in that one or more precalculated key
15 pairs are stored in a memory of the first device, the or each key pair comprising a session key and an encrypted version of this session key prepared using a transport key, the encrypted value of the session key being subsequently communicated to the second device which decrypts this value using an equivalent transport key stored in its memory such that data communicated from at least the second to the first device
20 may thereafter be encrypted and decrypted by the session key in the respective devices.

Unlike the Digco system described above, the use of a precalculated stored pair of values avoids the necessity of having to provide an encryption algorithm within the
25 first device (e.g. the decoder) to encrypt an internally generated session key. As a consequence, the algorithm chosen to encrypt the session key need not be limited to a public/private key algorithm but may correspond to a symmetric type algorithm if desired. Nevertheless, as will be understood, the present invention may also be implemented using public/private key algorithms to encrypt the session key, as will
30 be discussed in further detail below.

Advantageously, a plurality of key pairs are stored in the memory of the first device,

the first device selecting and processing one or more session keys to generate a definitive session key and communicating the associated encrypted value or values to the second device for decryption and processing by the second device to generate the definitive session key.

5

The provision of a plurality of key pairs within the first device enables the first device to choose and define a different definitive session key for each communication session. In one embodiment, a subset of a plurality of stored session keys is chosen by the first device to generate the definitive session key, the associated encrypted values of these subset session keys being communicated to the second device for decryption and processing.

10

Depending on the type of operation used, the resulting definitive session key may be dependent on the order of combination of the chosen session keys. In such an embodiment, this order information is communicated to the second device to enable the second device to correctly generate the definitive session key using the associated encrypted values.

15

For example, an initial session key value known to both the first and second devices may be repeatedly encrypted in both devices by an ordered sequence of session keys using an encryption algorithm sensitive to the order of encryption, such as the DES symmetric algorithm.

20

Of course, where the first device is using a selected subset of keys to generate the definitive session key, it may not be necessary to also use an order dependent algorithm to generate a changeable definitive session key and the keys may be combined, for example, using a simple arithmetical operation.

25

In one advantageous embodiment, the one or more precalculated key pair values may be selected from a larger set of precalculated key pairs prior to storage in the first device. For example, the operator or system manager may communicate a large number of precalculated key pairs to the manufacturer of the first device, the device

30

manufacturer thereafter selecting at random the key pairs to be stored in a given device.

In this way, the key pair or pairs embedded in the first device will be unique to that
5 device, or at least quasi-unique, thereby increasing the level of security for the system. Furthermore, the entity responsible for manufacture of the device need not possess the algorithm or keys used to prepare the encrypted session key values but may be simply supplied with a table of key pairs.

10 Preferably, the encrypted key value or values communicated to the second device also include a signature value that may be read by the second device to verify the authenticity of the communicated value.

Such a signature value can be generated and verified in accordance with a
15 conventional signature system, for example using combination of hash and public/private key algorithms such as MD5 and RSA, this signature being appended to the key pair values stored in the first device.

Conveniently, the signature value can also be precalculated at the time of calculation
20 of the encrypted key value and thereafter stored in the first device.

In a particularly preferred embodiment, the algorithm and transport key used to
encrypt and decrypt the session key or keys correspond to a symmetric algorithm and
associated symmetric key. The use of a symmetric algorithm enables an increase in
25 the processing time necessary for the second device to decrypt the session key in comparison with an operation using a public/private key algorithm.

Whilst one of the advantages of the present invention lies in the adaptability of the
present system to use a symmetric algorithm, it will be appreciated that this is not
30 obligatory. For example, in an alternative embodiment, the session key or keys may be encrypted by a public key prior to storage in the first device and decrypted by an equivalent private key within the second device.

Further preferably, the encryption algorithm used with the session key to encrypt and decrypt data communicated between the first and second device (or vice versa) corresponds to a symmetric algorithm. The choice of algorithm used may depend on the system requirements such as the need to have bidirectional communication between the devices.

Suitable symmetric algorithms may include DES or even an appropriate proprietary algorithm. Suitable public/private key algorithms may comprise RSA or other similar algorithms.

As mentioned above, the present invention is particularly applicable to the field of digital television and, in one preferred embodiment, the first device corresponds to a decoder and the second device to a portable security module (or vice versa).

The portable security module may conveniently comprise a smart card. If so, the data encrypted with the session key may correspond to simple control word information used by the decoder to descramble broadcast data.

The same principle may also be applied to the case where the descrambling unit in the decoder is implemented as a detachable conditional access module or CAM, broadcast data being descrambled in the conditional access module and communicated to the decoder.

In this embodiment, the first device may thus correspond to a decoder and the second device to a detachable conditional access module. If so, the data encrypted with the session key will normally correspond to the data descrambled by the conditional access module e.g. the broadcast programme itself.

In a conditional access module implementation, a smart card may also form part of the system, this card being inserted in the conditional access module to decrypt the control word, which is then passed to the conditional access module to permit descrambling of the broadcast programme. If so, the first device may then correspond to a

conditional access module, the second device to a smart card and the data encrypted with the session key to control word data.

5 Within the field of digital television, the invention may also be applied to the communication of data between a decoder and other devices, such as a television or video recorder. In particular, in one embodiment, the first device corresponds to a first decoder and the second device to a second decoder.

10 In households possessing a first and second decoder, there are often a number of problems associated with maintaining communication between a first or "master" decoder and a second "slave" decoder. The use of a secure encrypted link to communicate audiovisual data, control word data, or even data relating to current subscription rights and exploitation keys, may prove useful in this context.

15 In yet a further realisation, the present invention may be applied to home network system, the first and second devices corresponding to first and second consumer electronic devices adapted to transfer data via a communication link (e.g. radio, PLC, infra-red etc.).

20 The above embodiments have been described in relation to a method of encryption of data. Viewed from another aspect, the invention may equally be applied to first and second devices adapted to carry out such a method.

25 Another aspect of the present invention provides a system for providing secure communication of data between first and second devices, said first device comprising a memory for storing at least one precalculated key pair comprising a session key and an encrypted version of the session key prepared using a transport key, and communication means, such as a communication link, for communicating the encrypted version of the session key to said second device, said second device
30 comprising a memory for storing an equivalent transport key, decryption means, such as a processor, for decrypting said encrypted version of the session key using said equivalent transport key, and means, such as the processor, for encrypting data to be

communicated to said first device using said session key.

Features described above relating to method aspects of the present invention can also be applied to device or system aspects, and vice versa.

5

As used above, the terms "portable security module", "smart card" and "conditional access module" may be interpreted in their broadest sense as applying to any portable microprocessor and/or memory based card capable of carrying out the described functions.

10

As particular examples of such devices, a smart card may correspond to a card device constructed in accordance with the known international standards ISO 7816-1, 7816-2 and 7816-3 whilst the conditional access module may be implemented as a PCMCIA or PC card corresponding to the standards fixed by the PCMCIA group. Other physical shapes and forms are of course possible.

15

The terms "scrambled" and "encrypted" and "control word" and "key" have been used at various parts in the text for the purpose of clarity of language. However, it will be understood that no fundamental distinction is to be made between "scrambled data" and "encrypted data" or between a "control word" and a "key".

20

Similarly, unless obligatory in view of the context stated or unless otherwise specified, no limitation to either symmetric or public/private algorithms is to be inferred for a given encryption and/or decryption process. In the same way, whilst the matching keys used in encrypting and decrypting information may be referred to by the same name (e.g. "transport key", "session key") it is to be understood that these need not be numerically identical keys as long as they fulfil their functions. For example, the corresponding public and private keys used to encrypt and decrypt data will normally possess numerically different values.

25

30

The term "receiver/decoder" or "decoder" as used herein may connote a receiver for receiving either encoded or non-encoded signals, for example, television and/or radio

signals, which may be broadcast or transmitted by any appropriate means. Embodiments of such decoders may also include a decoder integral with the receiver for decoding the received signals, for example, in a “set-top box”, a decoder functioning in combination with a physically separate receiver, or such a decoder including additional functions, such as a web browser, integrated with other devices such as a video recorder or a television.

As used herein, the term “digital transmission system” includes any transmission system for transmitting or broadcasting for example primarily audiovisual or multimedia digital data. Whilst the present invention is particularly applicable to a broadcast digital television system, the invention may also be applicable to a fixed telecommunications network for multimedia internet applications, to a closed circuit television, and so on.

As used herein, the term “digital television system” includes for example any satellite, terrestrial, cable and other system.

There will now be described, by way of example only, a number of embodiments of the invention, with reference to the following figures, in which:

20

Figure 1 shows by way of background the overall architecture of a digital TV system;

Figure 2 shows the architecture of the conditional access system of Figure 1;

Figure 3 shows a method of encryption of data between a smart card and a decoder according to this embodiment of the invention;

Figure 4 shows the generation of a session key in a decoder operating according to the embodiment of Figure 3; and

30

Figure 5 shows the steps in the preparation of a session key in a smart card interfacing with the decoder of Figure 4.

The present invention describes a method of encryption of data, in particular but not exclusively applicable to the encryption of data across the interface between a portable security module and decoder in a digital television system. By way of background, the architecture of a known digital television system will now be described.

5

Digital Television System

An overview of a digital television system 1 is shown in Figure 1 comprising a broadcast system 2 which uses the MPEG-2 compression system to transmit compressed digital signals. In more detail, an MPEG-2 compressor 3 in a broadcast centre receives a digital signal stream (for example a stream of audio or video signals). The compressor 3 is connected to a multiplexer and scrambler 4 by linkage 5. The multiplexer 4 receives a plurality of further input signals, assembles one or more transport streams and transmits compressed digital signals to a transmitter 6 of the broadcast centre via linkage 7, which can of course take a wide variety of forms including telecom links.

10
15

The transmitter 6 transmits electromagnetic signals via uplink 8 towards a satellite transponder 9, where they are electronically processed and broadcast via a notional downlink 10 to earth receiver 11, conventionally in the form of a dish owned or rented by the end user. The signals received by receiver 11 are transmitted to an integrated receiver/decoder 12 owned or rented by the end user and connected to the end user's television set 13. The receiver/decoder 12 decodes the compressed MPEG-2 signal into a television signal for the television set 13.

20
25

A conditional access system 20 is connected to the multiplexer 4 and the receiver/decoder 12, and is located partly in the broadcast centre and partly in the decoder. It enables the end user to access digital television broadcasts from one or more broadcast suppliers. A portable security module in the form of a smartcard capable of decrypting messages relating to broadcast programmes or data can be inserted into the receiver/decoder 12.

30

An interactive system 17, also connected to the multiplexer 4 and the receiver/decoder 12 and again located partly in the broadcast centre and partly in the decoder, may be provided to enable the end user to interact with various applications via a modemmed back channel 16.

5

The conditional access system 20 will now be described in more detail. With reference to Figure 2, in overview the conditional access system 20 includes a Subscriber Authorization System (SAS) 21. The SAS 21 is connected to one or more Subscriber Management Systems (SMS) 22, one SMS for each broadcast supplier, for example by a respective TCP-IP linkage 23 (although other types of linkage could alternatively be used). Alternatively, one SMS could be shared between two broadcast suppliers, or one supplier could use two SMSs, and so on.

10

First encrypting units in the form of ciphering units 24 utilising "mother" smartcards 25 are connected to the SAS by linkage 26. Second encrypting units again in the form of ciphering units 27 utilising mother smartcards 28 are connected to the multiplexer 4 by linkage 29. The receiver/decoder 12 receives a portable security module, for example in the form of "daughter" smartcard 30. It is connected directly to the SAS 21 by Communications Servers 31 via the modemmed back channel 16. The SAS 20 sends, amongst other things, subscription rights to the daughter smartcard on request.

15

20

The smartcards contain the secrets of one or more commercial operators. The "mother" smartcard encrypts different kinds of messages and the "daughter" smartcards decrypt the messages, if they have the rights to do so.

25

The first and second ciphering units 24 and 27 comprise a rack, an electronic VME card with software stored on an EEPROM, up to 20 electronic cards and one smartcard 25 and 28 respectively, for each electronic card, one card 28 for encrypting the ECMs and one card 25 for encrypting the EMMs.

30

The operation of the conditional access system 20 of the digital television system will now be described in more detail with reference to the various components of the

television system 2 and the conditional access system 20.

Multiplexer and Scrambler

- 5 With reference to Figures 1 and 2, in the broadcast centre, the digital audio or video signal is first compressed (or bit rate reduced), using the MPEG-2 compressor 3. This compressed signal is then transmitted to the multiplexer and scrambler 4 via the linkage 5 in order to be multiplexed with other data, such as other compressed data.
- 10 The scrambler generates a control word used in the scrambling process and included in the MPEG-2 stream in the multiplexer. The control word is generated internally and enables the end user's integrated receiver/decoder 12 to descramble the programme.
- 15 Access criteria, indicating how the programme is commercialised, are also added to the MPEG-2 stream. The programme may be commercialised in either one of a number of "subscription" modes and/or one of a number of "Pay Per View" (PPV) modes or events. In the subscription mode, the end user subscribes to one or more commercial offers, or "bouquets", thus getting the rights to watch every channel inside
- 20 those bouquets. In the preferred embodiment, up to 960 commercial offers may be selected from a bouquet of channels.

In the Pay Per View mode, the end user is provided with the capability to purchase events as he wishes. This can be achieved by either pre-booking the event in advance

25 ("pre-book mode"), or by purchasing the event as soon as it is broadcast ("impulse mode"). In the preferred embodiment, all users are subscribers, whether or not they watch in subscription or PPV mode, but of course PPV viewers need not necessarily be subscribers.

30 **Entitlement Control Messages**

Both the control word and the access criteria are used to build an Entitlement Control

Message (ECM). This is a message sent in relation with a scrambled program; the message contains a control word (which allows for the descrambling of the program) and the access criteria of the broadcast program. The access criteria and control word are transmitted to the second encrypting unit 27 via the linkage 29. In this unit, an
5 ECM is generated, encrypted and transmitted on to the multiplexer and scrambler 4. During a broadcast transmission, the control word typically changes every few seconds, and so ECMs are also periodically transmitted to enable the changing control word to be descrambled. For redundancy purposes, each ECM typically includes two control words; the present control word and the next control word.

10

Each service broadcast by a broadcast supplier in a data stream comprises a number of distinct components; for example a television programme includes a video component, an audio component, a sub-title component and so on. Each of these components of a service is individually scrambled and encrypted for subsequent
15 broadcast to the transponder 9. In respect of each scrambled component of the service, a separate ECM is required. Alternatively, a single ECM may be required for all of the scrambled components of a service. Multiple ECMs are also generated in the case where multiple conditional access systems control access to the same transmitted program.

20

Entitlement Management Messages (EMMs)

The EMM is a message dedicated to an individual end user (subscriber), or a group of end users. Each group may contain a given number of end users. This organisation
25 as a group aims at optimising the bandwidth; that is, access to one group can permit the reaching of a great number of end users.

Various specific types of EMM can be used. Individual EMMs are dedicated to individual subscribers, and are typically used in the provision of Pay Per View
30 services; these contain the group identifier and the position of the subscriber in that group.

Group subscription EMMs are dedicated to groups of, say, 256 individual users, and are typically used in the administration of some subscription services. This EMM has a group identifier and a subscribers' group bitmap.

5 Audience EMMs are dedicated to entire audiences, and might for example be used by a particular operator to provide certain free services. An "audience" is the totality of subscribers having smartcards which bear the same conditional access system identifier (CA ID). Finally, a "unique" EMM is addressed to the unique identifier of the smartcard.

10

EMMs may be generated by the various operators to control access to rights associated with the programs transmitted by the operators as outlined above. EMMs may also be generated by the conditional access system manager to configure aspects of the conditional access system in general.

15

The term EMM is also often used to describe specific configuration type messages communicated between the decoder and other elements of the system and, for example, will be used later in this application to refer to a specific message passed from the decoder to a smart card.

20

Subscriber Management System (SMS)

A Subscriber Management System (SMS) 22 includes a database 32 which manages, amongst others, all of the end user files, commercial offers, subscriptions, PPV details, 25 and data regarding end user consumption and authorization. The SMS may be physically remote from the SAS.

Each SMS 22 transmits messages to the SAS 21 via respective linkage 23 which imply modifications to or creations of Entitlement Management Messages (EMMs) to be 30 transmitted to end users.

The SMS 22 also transmits messages to the SAS 21 which imply no modifications or

creations of EMMs but imply only a change in an end user's state (relating to the authorization granted to the end user when ordering products or to the amount that the end user will be charged).

- 5 The SAS 21 sends messages (typically requesting information such as call-back information or billing information) to the SMS 22, so that it will be apparent that communication between the two is two-way.

Subscriber Authorization System (SAS)

10

The messages generated by the SMS 22 are passed via linkage 23 to the Subscriber Authorization System (SAS) 21, which in turn generates messages acknowledging receipt of the messages generated by the SMS 21 and passes these acknowledgements to the SMS 22.

15

- In overview the SAS comprises a Subscription Chain area to give rights for subscription mode and to renew the rights automatically each month, a Pay Per View Chain area to give rights for PPV events, and an EMM Injector for passing EMMs created by the Subscription and PPV chain areas to the multiplexer and scrambler 4, and hence to feed the MPEG stream with EMMs. If other rights are to be granted, such as Pay Per File (PPF) rights in the case of downloading computer software to a user's Personal Computer, other similar areas are also provided.
- 20

- One function of the SAS 21 is to manage the access rights to television programmes, available as commercial offers in subscription mode or sold as PPV events according to different modes of commercialisation (pre-book mode, impulse mode). The SAS 21, according to those rights and to information received from the SMS 22, generates EMMs for the subscriber.
- 25

- The EMMs are passed to the Cipherring Unit (CU) 24 for cipherring with respect to the management and exploitation keys. The CU completes the signature on the EMM and passes the EMM back to a Message Generator (MG) in the SAS 21, where a header
- 30

-16-

is added. The EMMs are passed to a Message Emitter (ME) as complete EMMs. The Message Generator determines the broadcast start and stop time and the rate of emission of the EMMs, and passes these as appropriate directions along with the EMMs to the Message Emitter. The MG only generates a given EMM once; it is the ME which performs cyclic transmission of the EMMs.

On generation of an EMM, the MG assigns a unique identifier to the EMM. When the MG passes the EMM to the ME, it also passes the EMM ID. This enables identification of a particular EMM at both the MG and the ME.

Programme Transmission

The multiplexer 4 receives electrical signals comprising encrypted EMMs from the SAS 21, encrypted ECMs from the second encrypting unit 27 and compressed programmes from the compressor 3. The multiplexer 4 scrambles the programmes and sends the scrambled programmes, the encrypted EMMs and the encrypted ECMs to a transmitter 6 of the broadcast centre via the linkage 7. The transmitter 6 transmits electromagnetic signals towards the satellite transponder 9 via uplink 8.

Programme Reception

The satellite transponder 9 receives and processes the electromagnetic signals transmitted by the transmitter 6 and transmits the signals on to the earth receiver 11, conventionally in the form of a dish owned or rented by the end user, via downlink 10. The signals received by receiver 11 are transmitted to the integrated receiver/decoder 12 owned or rented by the end user and connected to the end user's television set 13. The receiver/decoder 12 demultiplexes the signals to obtain scrambled programmes with encrypted EMMs and encrypted ECMs.

If the programme is not scrambled, that is, no ECM has been transmitted with the MPEG-2 stream, the receiver/decoder 12 decompresses the data and transforms the signal into a video signal for transmission to television set 13.

If the programme is scrambled, the receiver/decoder 12 extracts the corresponding ECM from the MPEG-2 stream and passes the ECM to the "daughter" smartcard 30 of the end user. This slots into a housing in the receiver/decoder 12. The daughter smartcard 30 controls whether the end user has the right to decrypt the ECM and to
5 access the programme. If the end user does have the rights, the ECM is decrypted within the smart card and the control word extracted.

Thereafter the smart card then communicates the control word to the decoder 12 which then descrambles the programme using this control word. In most conventional
10 systems, the control word is communicated across the smart card interface in a clear or non-encrypted form, leading to the problems of security described in the introduction of the present application. After descrambling by the decoder, the MPEG-2 stream is decompressed and translated into a video signal for onward transmission to television set 13.

15 In the system described above, the descrambling of the MPEG data is carried out within the decoder using the control word information communicated to the decoder from the smart card. In other systems, the descrambling circuitry may be implemented in a detachable conditional access module or CAM, commonly embodied in the form
20 of a PCMCIA or PC card insertable in a socket in the decoder.

The CAM module may itself further include a slot to receive a smart card. In such systems, control word data is decrypted in the smart card communicated to the CAM module which then descrambles the scrambled MPEG data stream to supply the
25 decoder with a clear MPEG stream for decompression and subsequent display.

In this type of system, sensitive data may be passed between the smart card and CAM (control word data) and/or between the CAM and decoder (descrambled MPEG data) and problems of security may arise at either of these interfaces.

30

Data Encryption across an Interface

Referring to Figure 3, there will now be described a method of data encryption as applied to the control word data communicated between a smart card and a decoder in one of the simplest embodiments of this invention. However, the same principles may be applied to the encryption of control word data between a smart card and a CAM, audiovisual MPEG data between a CAM and a decoder, or indeed any type of data between two such devices.

In accordance with the present invention, a set of key pairs is stored in a non-volatile memory of the decoder e.g. a FLASH memory. Each key pair corresponds to a key value in clear form and an encrypted version of the key. As will be described, the encrypted version of the key will be eventually communicated in an EMM message sent to a smart card inserted in the decoder.

Thus, within the decoder a set of EMM message/key pairs are stored as follows:

15	n	EMM (19 octets)	Key (8 octets)
	1	EMM(1)	Key(1)
	2	EMM(2)	Key(2)
20	3	EMM(3)	Key(3)
	.	.	.
	.	.	.
	.	.	.
25	16	EMM(16)	Key(16)

The encrypted value of the key stored in the EMM is calculated external of the decoder using an encryption algorithm not present in the decoder. In the present example the key values Key(1), Key(2) etc. correspond to symmetric keys to be used with a symmetric encryption algorithm such as DES.

The encryption algorithm used to prepare the encrypted DES key values contained with the stored EMM messages may also correspond to a symmetric encryption algorithm. For increased security, a proprietary symmetric algorithm (PSA) different from DES will be used to prepare the encrypted values, although in another

embodiment DES may also be used to encrypt the key values.

In addition to the encrypted value of the associated key, the EMM message may also include a signature value associated with the message and prepared as per any conventional signature preparation method. For example, a message may be subject
5 to a hash function such as MD5 followed by encryption of the hash value by a private key of private/public key algorithm such as RSA. Verification of the signature may then be carried out at the point of reception using a MD5 algorithm and the corresponding public key of the private/public key pair.

10

The EMM message will additionally include a standard smart card header element (as defined by the international standard ISO 7816-3) to place the message in a format necessary to permit it to be read by a smart card. An EMM associated with an 8 byte key will therefore typically have the following structure:

15

Header	5 bytes
Encrypted key	10 bytes
Signature	9 bytes

20 In the present embodiment a set of 16 key/message pairs are implanted in the memory of the decoder. Alternative embodiments are equally possible using more or less key/message pairs and the invention may even be implemented using a single key/message pair. Whilst it may be envisaged that all decoders are equipped with the same key/message pairs it is preferred for security reasons that each decoder has a
25 unique set of key/message pairs. In implementing this embodiment, an operator may supply to a decoder manufacturer a set of ten thousand or more key/message pairs, the decoder manufacturer taking a random selection of 16 pairs during the personalisation of each decoder.

30 In order to increase the security, a different subset of the message/key pairs stored in the decoder will be used during each session. A session may be defined as corresponding to each time the decoder is switched on and off, or each time the

decoder changes channel, for example.

Referring to Figure 3, a random number generator 40 within the decoder selects 8 out of the 16 message/key pairs to be used in that session. The 8 selected EMM messages
5 41 of the pairs are then communicated to the smart card 30 to be verified and decrypted and processed as shown at 42 and 43 to obtain the appropriate session key (see below). The same key generation operation is carried out within the decoder at 43 using the corresponding key values of the pairs so as to obtain the same session key value.

10

The generation of the session key within the decoder will now be described with reference to Figure 4.

A base session key value KeyS Initial shown at 44 and constant for all decoders is
15 encrypted at 45 by the first key 46 of the subset chosen by the random generator 40. The resulting value is then encrypted at 47 using the second key 48 of the session subset and the operation repeated just until the last encryption operation 49 carried out with the last key 50 of the subset so as to obtain the final session key value shown at 51.

20

The initial session key value KeyS Initial can be a universal value present in all decoders and smart cards, a value linked to a specific decoder/smart card pair or even a value generated at the start of each session in the decoder and thereafter communicated to the smart card.

25

In the example given above, the session key is prepared by a sequence of repeated operations on the KeyS Initial using the DES algorithm and the selected keys 46, 48, 50 etc. In the case of the DES algorithm, the order in which the keys are applied is important and must be respected to produce the same key each time.

30

However, whilst the session key S is itself a numerical value that will be used as a DES key in the subsequent decryption operation (see below), the steps used to

generate this key value need not correspond to DES encryption steps. Instead, the subset of keys chosen by the random number generator may be combined together in any number of ways to arise at a suitable session key value KeyS Final. For example, the keys may be combined using a sequence of simple arithmetic operations.

5 Depending on the method chosen, it may not be necessary that the order of the steps in the preparation of the KeyS be respected in order to regenerate the same key.

Referring now to Figure 5, the decryption and processing operations 42 and 43 carried out in the smart card 30 to generate the session key used by the smart card will now be described.

10

Upon insertion of the smart card in the decoder, the subset of EMM messages matching the selected key values are sent to the smart card. Authentication of each EMM messages is first carried out with reference to the attached signature value, using for example an MD5/RSA type process as described above. For simplicity, this step has been omitted from Figure 5.

15

The first EMM message 60 is then decrypted at 61 using a transport key 59 embedded in a secure and non-readable manner within the smart card. As mentioned above, for security reasons the algorithm used in the decryption 61 of the EMM message may correspond to a proprietary security algorithm PSA known only to the operator responsible for preparation of the message/key pairs used in the decoder and the personalisation of the smart card.

20

The transport key KeyT shown at 59 may be a key value common to all smart cards in the system or unique to one such card. The use of a unique key value KeyT requires that the message/key table stored in the decoder be prepared with the same key as that in the card, such that a decoder and card will be irreversibly linked together. In practice, this may not be desirable.

25

30

A similar decryption operation using the transport key 59 is then carried out at 62 on the next EMM message 63 in the series and 50 on until the last decryption operation

64 on the final EMM message 65.

5 In the present embodiment, encryption of each of the EMM messages 60, 63, 65 produces keys 46, 48, 50 identical to those associated in the message/key table present in the decoder and used for generation of the session key as described previously. For this reason, the same reference numbers have been used for these keys and for the key generation operation 43 also carried out in the decoder. Similarly, the same initial session key 44 present in the decoder is also stored in the smart card.

10 The initial session key KeyS Initial shown at 44 is then encrypted at 45 by the first key 46, the result re-encrypted at 47 by the second key 48 and so on until the final encryption step carried out at 49 using the last key 50 in the series so as to obtain the final session key at 51.

15 Both the decoder and smart card now possess the same session key KeyS which may thereafter be used in encrypting and decrypting data passed in either direction between the two devices.

20 Referring back to Figure 3, the smart card 30 receives an encrypted ECM message containing the control word necessary for descrambling an associated segment of MPEG audiovisual or other data. The smart card decrypts the ECM at 71 to obtain the control word value CW.

25 In passing, we note that the algorithm used to encrypt ECM messages for a user may conveniently correspond to the Proprietary Security Algorithm used for decryption of the EMM messages received from the smart card as described above.

30 The decrypted control word is then re-encrypted at 72 using the session key KeyS and the encrypted control word value $f(CW)$ transmitted over the decoder/smart card interface as shown. The encrypted value $f(CW)$ is then decrypted at 73 using the session key KeyS held in the decoder and the clear value of the control word CW obtained at 74.

As the session key is symmetric, it may equally be used in the encryption of data transmitted from the decoder to the smart card. Furthermore, the data transmitted from the smart card to the decoder may be data other than simple control word data.

- 5 As mentioned above, the same principle may be applied across all interfaces in a system comprising a decoder in which a detachable CAM module is inserted (decoder/CAM interface, CAM/smart card interface etc.). Similarly, the same principle may be applied in the case of a portable module (either a CAM type module or a smart card) inserted in other devices such as a television or video recorder.

10

- In fact, the above method of setting up an encrypted communication channel may be applied to any pair of devices where security of data communication is required. In particular, the same principle may be applied in a home network system where multiple consumer devices (television, video, PC, decoder etc.) transfer data such as
- 15 audiovisual data or computer files via a communication link. This may be an RF link, an infrared link, a dedicated bus, a power line connection etc. For example, it may be desired to transmit control word in other data in an encrypted form between a decoder and a television or between a master decoder and a slave decoder in the same household.

20

Other examples of systems of this type where a secure communication link would be desirable will also be apparent to the reader.

CLAIMS

1. A method of encryption of data communicated between a first and second device, wherein at least one precalculated key pair is stored in a memory of the first device,
5 said at least one key pair comprising a session key and an encrypted version of the session key prepared using a transport key, the encrypted version of the session key being subsequently communicated to the second device which decrypts the encrypted version using an equivalent transport key stored in its memory such that data
10 communicated from at least the second to the first device may thereafter be encrypted and decrypted by the session key in the respective devices.
2. A method as claimed in claim 1, in which a plurality of key pairs are stored in the memory of the first device, the first device selecting and processing at least one session key to generate a definitive session key and communicating the associated
15 encrypted version of said at least one session key to the second device for decryption and processing by the second device to generate the definitive session key.
3. A method as claimed in claim 2 in which a subset of a plurality of stored session keys is chosen by the first device to generate the definitive session key, the associated
20 encrypted versions of the subset of session keys being communicated to the second device for decryption and processing.
4. A method as claimed in claim 2 or 3, in which the order of combination of a plurality of session keys used to generate the definitive session key is communicated
25 from the first to the second device.
5. A method as claimed in claim 4 in which an initial session key value known to both the first and second devices is repeatedly encrypted in both devices by an ordered sequence of session keys using an encryption algorithm sensitive to the order of
30 encryption.
6. A method as claimed in any preceding claim in which said at least one

precalculated key pair is selected from a larger set of precalculated key pairs prior to being stored in the first device.

7. A method as claimed in any preceding claim in which the encrypted version of a
5 session key communicated to the second device also includes a signature value
readable by the second device to verify the authenticity of the encrypted version of the
session key.

8. A method as claimed in any preceding claim in which an algorithm and transport
10 key used to encrypt and decrypt a session key correspond to a symmetric algorithm
and associated symmetric key.

9. A method as claimed in any preceding claim in which an encryption algorithm
used with a session key to encrypt and decrypt data communicated between the first
15 and second device corresponds to a symmetric algorithm.

10. A method as claimed in any preceding claim, in which the first device is a
decoder.

20 11. A method as claimed in any preceding claim, in which the second device is a
portable security module.

12. A method as claimed in claim 11, in which the portable security module
corresponds to one of a smart card and a conditional access module.

25 13. A method as claimed in any of claims 1 to 9, in which the first device
corresponds to a conditional access module and the second device corresponds to a
smart card.

30 14. A method as claimed in any of claims 10 to 13, in which data encrypted and
decrypted with a session key corresponds to control word data.

15. A method as claimed in any of claims 10 to 13, in which data encrypted and decrypted with a session key corresponds to descrambled broadcast data.

16. A method as claimed in any of claims 1 to 9 in which the first and second device
5 correspond to a first and second decoder respectively.

17. A method as claimed in any of claims 1 to 9 as applied to a home network system, the first and second devices corresponding to first and second consumer electronic devices adapted to transfer data via a communication link.
10

18. A first device adapted to be used in a method as claimed in any of claims 1 to 17, the first device including a memory in which at least one precalculated key pair is stored, said at least one precalculated key pair comprising a session key and an encrypted version of this session key.
15

19. A second device adapted to be used in a method as claimed in any of claims 1 to 18 and with a first device as claimed in claim 18, the second device comprising a memory in which is stored a key and algorithm that are needed to decrypt the encrypted session key value stored in the memory of the first device.
20

20. A first and second device as claimed in claims 18 and 19, in which the first device corresponds to a decoder and the second device to a portable security module.

21. A system for providing secure communication of data between first and second
25 devices, said first device comprising a memory for storing at least one precalculated key pair comprising a session key and an encrypted version of the session key prepared using a transport key, and communication means for communicating the encrypted version of the session key to said second device, said second device comprising a memory for storing an equivalent transport key, decryption means for
30 decrypting said encrypted version of the session key using said equivalent transport key, and means for encrypting data to be communicated to said first device using said session key.

22. A system as claimed in claim 21, wherein the memory of the first device is adapted to store a plurality of key pairs, the first device comprising means for selecting and processing at least one session key to generate a definitive session key
5 said communication means being adapted to communicate the associated encrypted version of said at least one session key to the second device, said second device comprising means for processing said at least one session key to generate the definitive session key.
- 10 23. A system as claimed in claim 21 or 22, in which the encrypted version of a session key includes a signature value readable by the second device to verify the authenticity of the encrypted version of the session key.
- 15 24. A system as claimed in any of claims 21 to 23, in which an algorithm and transport key used to encrypt and decrypt a session key correspond to a symmetric algorithm and associated symmetric key.
- 20 25. A system as claimed in any of claims 21 to 24, in which an encryption algorithm used with a session key to encrypt and decrypt data communicated between the first and second device corresponds to a symmetric algorithm.
- 25 26. A system as claimed in any of claims 21 to 25, in which the first device is a decoder.
27. A system as claimed in any of claims 21 to 26, in which the second device is a portable security module.
28. A system as claimed in claim 27, in which the portable security module corresponds to one of a smart card and a conditional access module.
- 30 29. A system as claimed in any of claims 21 to 25, in which the first device corresponds to a conditional access module and the second device corresponds to a

smart card.

30. A system as claimed in any of claims 21 to 25 in which the first and second device correspond to a first and second decoder respectively.

5

31. A system as claimed in any of claims 21 to 25 as applied to a home network system, the first and second devices corresponding to first and second consumer electronic devices adapted to transfer data via a communication link.

10 32. A method of encryption of data communicated between a first and second device substantially as herein described.

33. A system for providing secure communication of data between first and second devices substantially as herein described.

15

FIG. 1

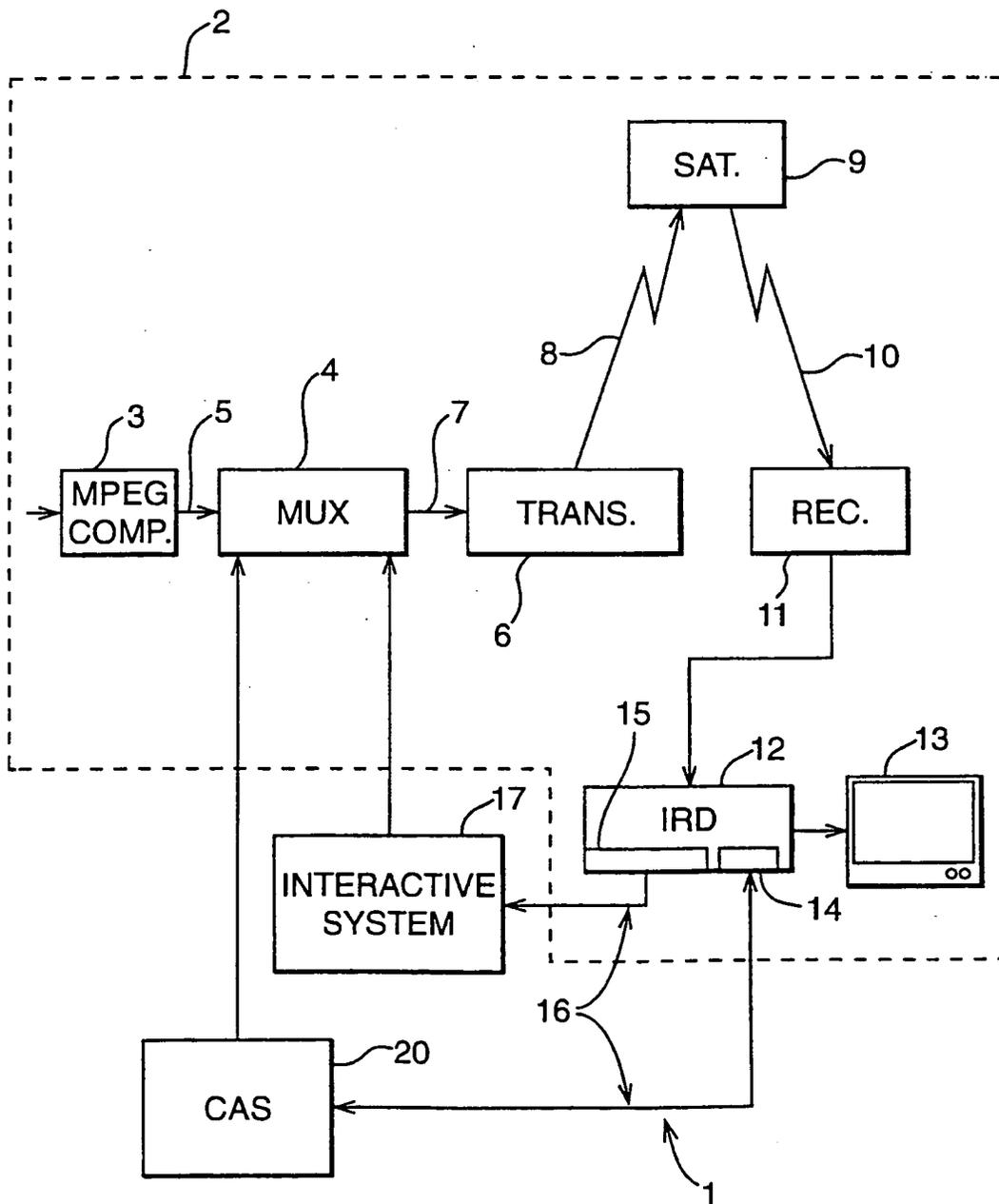


FIG. 2

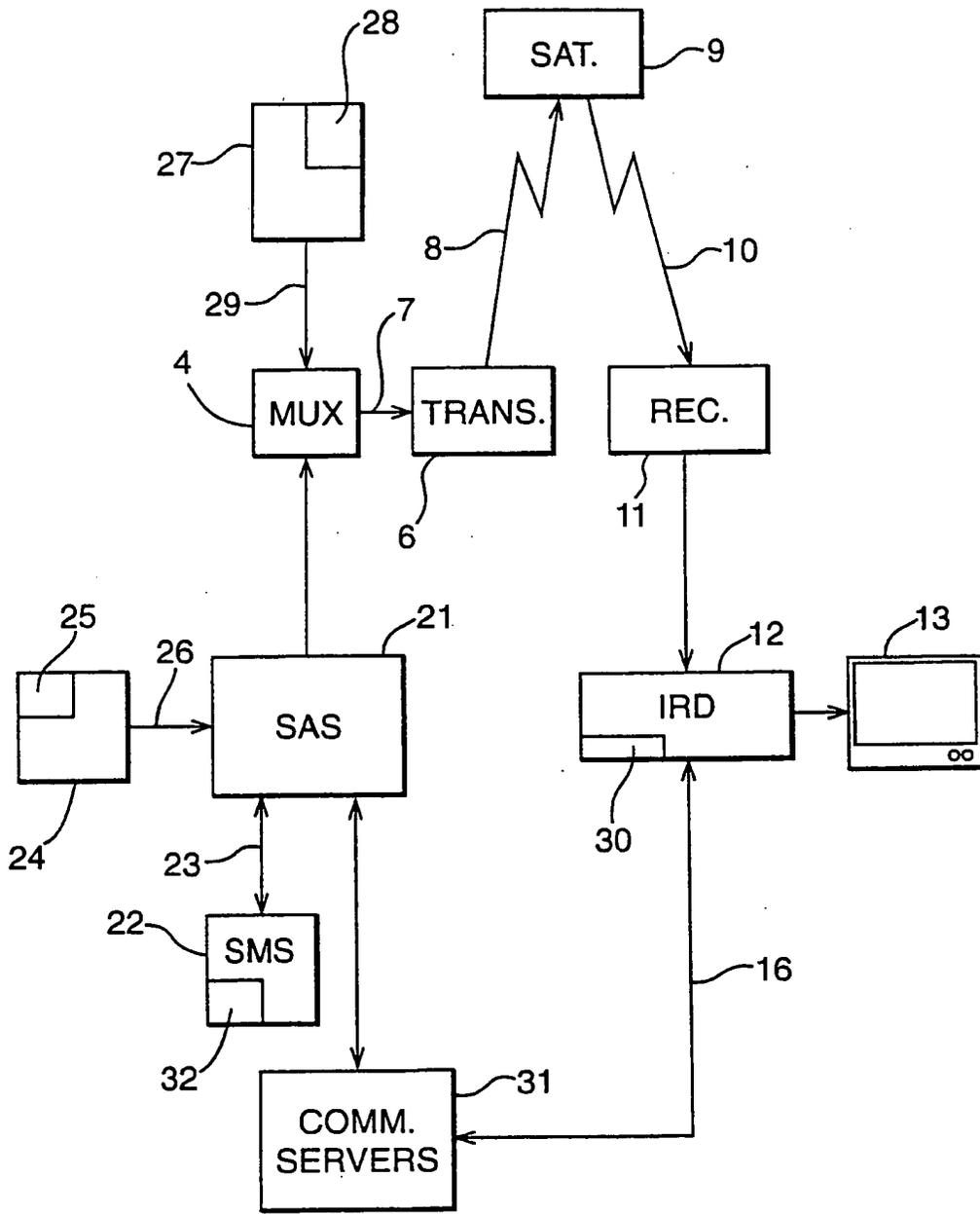
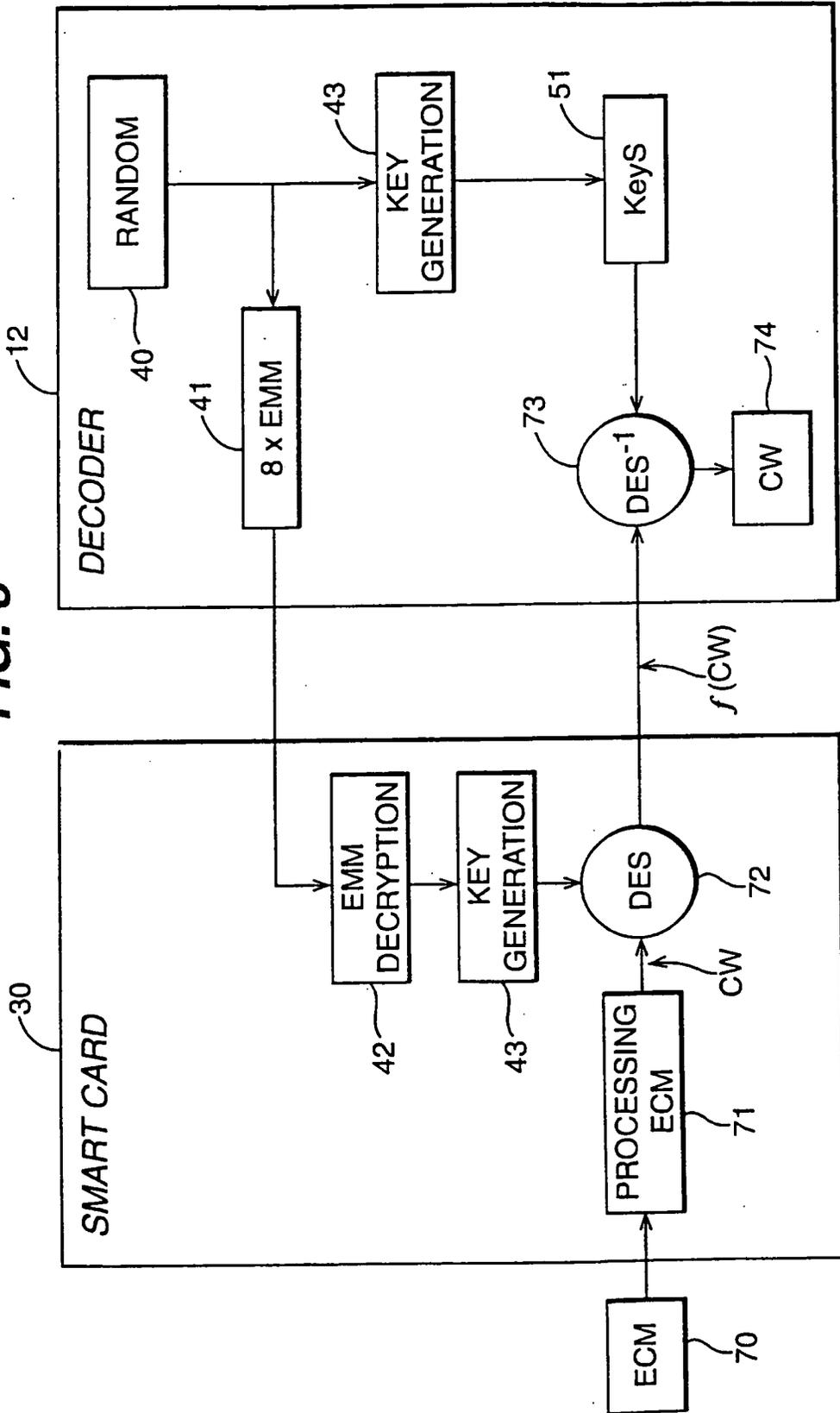
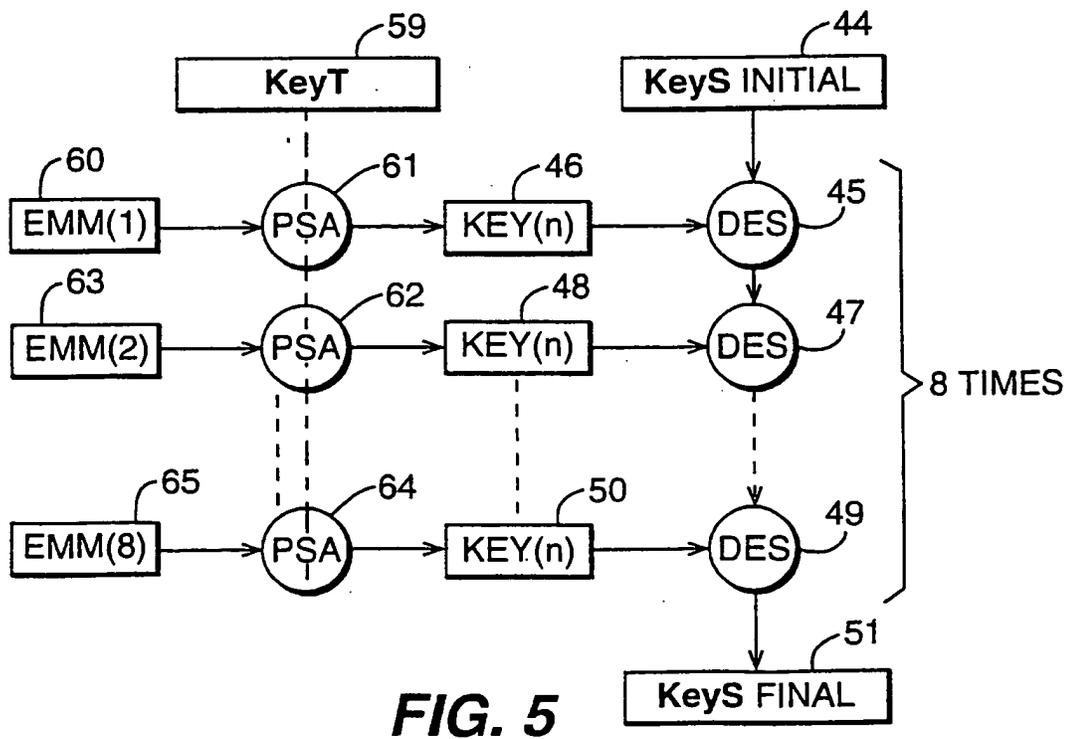
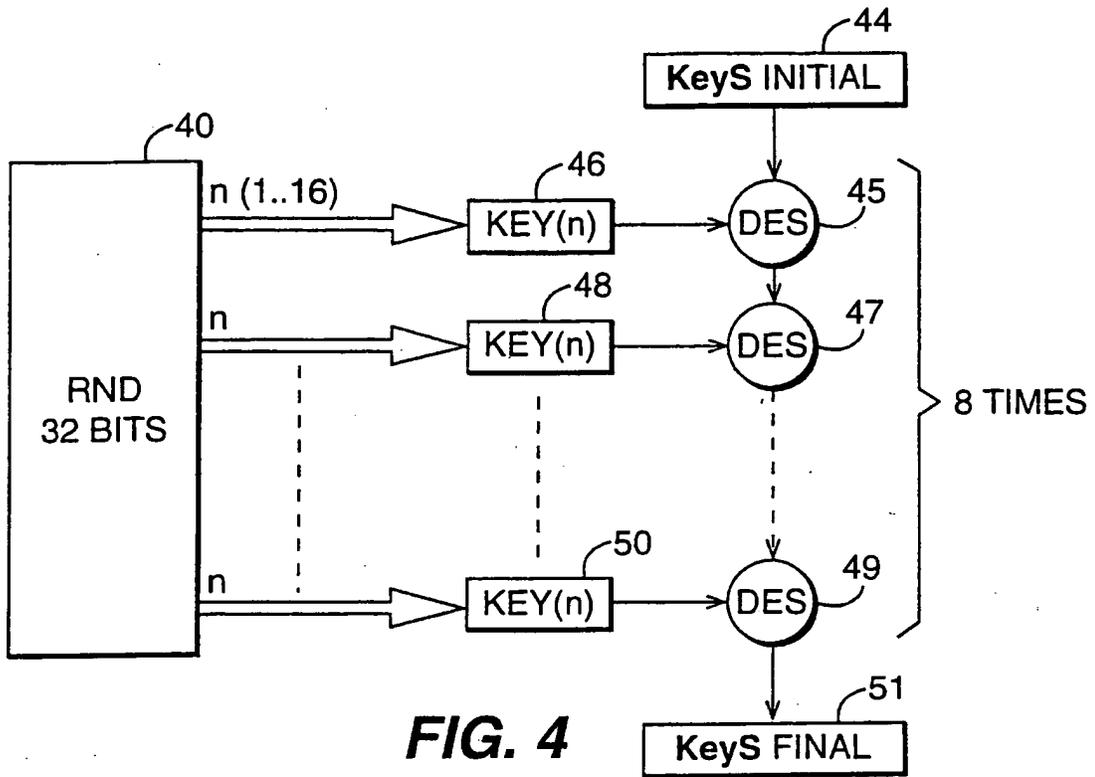


FIG. 3





INTERNATIONAL SEARCH REPORT

International Application No
PCT/IB 00/00163

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04N7/16 H04N7/167

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 817 485 A (THOMSON MULTIMEDIA SA) 7 January 1998 (1998-01-07) page 3, column 3, line 54 -page 5, column 8, line 11 figures 1-5	1, 2, 4, 10-15, 17, 19-22, 26-29
X	EP 0 723 371 A (THOMSON MULTIMEDIA SA) 24 July 1996 (1996-07-24) page 3, column 3, line 57 -page 5, column 7, line 8 figures 1-4	1, 2, 4, 10-15, 19-22, 26-29
	-/-	

Further documents are listed in the continuation of box C. Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
 "E" earlier document but published on or after the international filing date
 "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
 "O" document referring to an oral disclosure, use, exhibition or other means
 "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
 "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
 "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
 "&" document member of the same patent family

Date of the actual completion of the international search 31 May 2000	Date of mailing of the international search report 07/06/2000
--	--

Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Van der Zaal, R
--	---

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int'l Patent Application No
PCT/IB 00/00163

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>EBU PROJECT GROUP B/CA: "FUNCTIONAL MODEL OF A CONDITIONAL ACCESS SYSTEM" EBU REVIEW- TECHNICAL, no. 266, 21 December 1995 (1995-12-21), pages 64-77, XP000559450 Grand Saconnex, CH page 64, left-hand column, line 1 -page 72, right-hand column, line 29 figures 1-8</p>	1-33

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/IB 00/00163

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 0817485	A	07-01-1998	FR 2750554 A	02-01-1998
			CN 1171015 A	21-01-1998
			JP 10164052 A	19-06-1998
			US 6035038 A	07-03-2000
EP 0723371	A	24-07-1996	FR 2729521 A	19-07-1996
			JP 8307850 A	22-11-1996

**(WO/2000/062260) METHOD AND SYSTEM FOR ORDERING, LOADING AND USING ACCESS TICKETS**

Biblio. Data	Description	Claims	National Phase	Notices	Documents
--------------	-------------	--------	----------------	---------	-----------

Latest bibliographic data on file with the International Bureau

Publication Number: WO/2000/062260 **International Application No.:** PCT/CH1999/000142
Publication Date: 19.10.2000 **International Filing Date:** 07.04.1999
Chapter 2 Demand Filed: 22.04.2000

Int. Class.: G06Q 20/00 (2006.01), G07B 15/00 (2006.01), G07F 17/42 (2006.01), G07F 7/00 (2006.01), G07F 7/08 (2006.01)

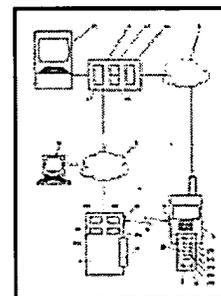
Applicants: SWISSCOM MOBILE AG [CH/CH]; Schwarztorstrasse 61 CH-3050 Bern (CH) (*All Except US*).
 RITTER, Rudolf [CH/CH]; Rossweidweg 8 CH-3052 Zollikofen (CH) (*US Only*).
 LAUPER, Eric [CH/CH]; Hochfeldstrasse 96 CH-3012 Bern (CH) (*US Only*).

Inventors: RITTER, Rudolf [CH/CH]; Rossweidweg 8 CH-3052 Zollikofen (CH).
 LAUPER, Eric [CH/CH]; Hochfeldstrasse 96 CH-3012 Bern (CH).

Agent: BOVARD AG; Optingenstrasse 16 CH-3000 Bern 25 (CH).

Title: METHOD AND SYSTEM FOR ORDERING, LOADING AND USING ACCESS TICKETS

Abstract: The invention relates to a method and a system for ordering, loading and using access tickets for the access to access-controlled service devices (3). Access tickets are ordered by a reservation centre (4) in said service device (3) by transmitting order information via an order channel. The order information comprises the telephone number of a mobile communications terminal (1). The ordered access tickets are transmitted to said terminal (1) via a mobile network (6) and are stored in a storage module (21) of the communications terminal (1). Data is exchanged between the storage module (21) and a reading device (31) of a service device (3) via a contactless interface (13). Decisions on the access permission for the user of said communications terminal (1) are made, e.g. in the reading device (31) or in the communications terminal (1), considering ticket information contained in said access ticket. Said information can be limited to a digitally signed ticket number or can contain data on the relevant service device.



Access for the user to the service device (3) is given or denied according to the decision and by means of an access device (32) that is connected to the reading device.

Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW.
 African Regional Intellectual Property Org. (ARIPO) (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW)
 Eurasian Patent Organization (EAPO) (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM)
 European Patent Office (EPO) (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE)
 African Intellectual Property Organization (OAPI) (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Publication Language: German (DE)

Filing Language: German (DE)

(19) World Intellectual Property Organization
International Bureau



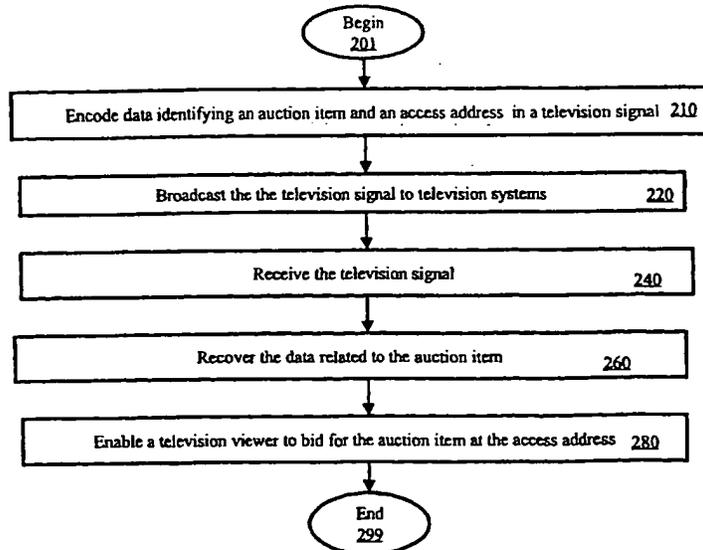
(43) International Publication Date
11 January 2001 (11.01.2001)

PCT

(10) International Publication Number
WO 01/03044 A1

- (51) International Patent Classification⁷: G06F 17/60
- (21) International Application Number: PCT/US00/18510
- (22) International Filing Date: 6 July 2000 (06.07.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
09/347,391 6 July 1999 (06.07.1999) US
- (71) Applicant (for all designated States except US): TRANSCAST INTERNATIONAL, INC. [US/US]; Regency Plaza, 2350 Mission College Blvd., Suite 190, Santa Clara, CA 95054 (US).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): NARAYAN, Kris [US/US]; 983 Sandalridge Court, Milpitas, CA 95035 (US).
- (74) Agent: THAPPETA, Narendra, Reddy; Law Firm of Naren Thappeta, 39899 Balentine Drive #119, Newark, CA 94560 (US).
- (81) Designated States (national): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:
— With international search report.
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: ENABLING VIEWERS OF TELEVISION SYSTEMS TO PARTICIPATE IN AUCTIONS



(57) Abstract: Enabling the viewers of television systems to participate in auctions. Data identifying an item (e.g., description of the auction item and a unique code) offered for sale in an auction and an access address (e.g., universal resource locator of a web site) may be encoded (210) in a television signal and broadcast (220) to various television systems. The data may be recovered (240, 260) by a transaction enabler which enables a viewer to bid for the auction item (280). Other information such as highest present bid price may also be encoded in the television signal and displayed for the viewer.

WO 01/03044 A1

ENABLING VIEWERS OF TELEVISION SYSTEMS TO PARTICIPATE IN AUCTIONS

Related Application

The present invention is related to co-pending U.S. Patent Application Entitled,
5 “Encoding Hot Spots in Television Signals”, Serial Number: 09/276,266, Filing Date: March
25, 1999, which is incorporated in its entirety into the present application.

Background of the Invention

Field of the Invention

The present invention relates to television systems, and more specifically to a method
10 and apparatus for using television signals to enable viewers of television systems to participate
in auctions.

Related Art

An auction generally refers to a process in which multiple parties are provided the
opportunity to bid for an offered item. The offered item can be a process or a service. In a
15 typical bidding process, an a seller offers an item, and a party (“bidder”) bids for the offered
item usually by specifying a price the party is willing to pay. The seller may specify the
minimum acceptable price and a time at which the auction closes.

Typically, an offered item is sold to the highest bidder (i.e., party specifying highest
price) in return for the specified highest price. However, criteria other than price (e.g., credit
20 worthiness) of the bidder may also be taken into consideration in determining the bidder to
whom to sell an offered item.

Central servers are known in the relevant arts which coordinate the bidding process.
For example, web site at URL of <http://www.ebay.com> enable sellers to offer products

according to various categories (e.g., sports memorabilia, computers), and a bidder may bid on the offered products by using a browser on the world-wide web as is well known in the relevant arts.

Organizations such as those providing the web sites to enable auctions are hereafter referred to as "service providers". Service providers often advertize on various other web sites so that users accessing ("surfing") these web sites may know about the general service. Typically, a user (viewer of the advertisement) can click on an advertisement to access the web sites providing the auction service.

However, these advertisements are typically targeted to the users surfing the world wide web, and may not target at least some of the viewers ("television viewers") of television systems. The television viewers constitutes a big segment of the auction market, and it is therefore desirable to enable television viewers to participate in the auctions.

Such participation may be particularly important as the viewers of a specific television program may be expected to be of certain 'profile', and certain items may be suitable for people of that profile. For example, a person watching Mr. Mark McGuire (a baseball player in United States baseball) hit a record breaking home run may be interested in purchasing a baseball bat signed personally by Mr. Mark McGuire. That is, the auction items can be targeted to the viewers of television programs.

At least for the above-stated reasons, what is needed is a method and apparatus for enabling viewers of television systems to participate in auctions.

Summary of the Invention

The present invention enables viewers ("television viewers") of television systems to participate in auctions. The auctions may be occurring on web sites on the Internet also. In

an embodiment of the invention, data describing an item ("auction item") available for bidding and an access address of a system at which a television viewer may bid are encoded in a television signal.

The user may submit a bid at a system (e.g., a web site) identified by the access address. In case the system is a web server, users ('surfers') of world-wide-web may also submit bids by accessing the web server on the world-wide web. Accordingly, the present invention may be used to draw television viewers to web-sites (e.g., www.ebay.com) dedicated to auctions also.

In an embodiment, the data is encoded in the non-display portion (e.g., vertical blanking interval) of the television signal. However, other portions of a television signal may also be used for encoding the data. Other information of interest to the viewer such as a minimum bid amount specified by a seller and the present maximum may also be encoded in the television signal, and displayed for viewer convenience.

A transaction enabler may recover the data encoded in the television signals, and display the information to the viewer. The viewer may conveniently bid on the auction items, for example, by specifying the bid price (offer) and clicking on a pre-specified portion of a displayed image.

The bid may be automatically sent to a server identified by the access address. In the alternative, the viewer may be first navigated to a web server specified by the access address, and the user may specify the bid price then. A unique code identifying the auction item may also be encoded in the television signal, and the code may be used to identify that the bid price relates to the auction item. In the alternative, the URL itself may contain such identification codes.

The transaction server may also provide updated information on a present highest bid. For example, an end time associated with the auction may be provided to the television viewer, and the viewer may check the present highest bid at a later time before the end time, and then decide whether to submit a bid. In addition, the transaction server may interact with the system providing the auction service, and provide periodic updates at viewer's option. As a result, a viewer may make an informed decision on whether to bid.

Therefore, the present invention enables a television viewer to participate in an auction by encoding in a television signal the data identifying an auction item and an access address.

The present invention enables television viewers to be drawn to web sites providing auction service by specifying the URL of the web site as the access address.

The present invention is useful for broadcasters as the broadcasters may facilitate the joining of additional bidders to a bidding process, and be compensated for such additions.

The present invention is useful for service providers providing auction service as the television viewers are drawn to bid for on-going auctions.

The present invention is useful for service providers providing auction service also because higher commissions may be charged for the auction items sold in accordance with the present invention.

The present invention is useful for television viewers as a television viewer may have non-intrusive access to information on auctions, and purchase the auction items by a convenient user interface.

The present invention is useful for sellers participating in auctions as the sellers may attain greater return for the auction items due to additional pool of bidders participating in accordance with the present invention.

Further features and advantages of the invention, as well as the structure and operation of various embodiments of the invention, are described in detail below with reference to the accompanying drawings. In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the corresponding reference number.

Brief Description of the Drawings

The present invention will be described with reference to the accompanying drawings, wherein:

Figure 1 is a block diagram illustrating an example environment in which the present invention can be implemented;

Figure 2 is a flow-chart illustrating a method in accordance with the present invention;

Figure 3 is a block diagram illustrating an example broadcast system which encodes data related to an auction item in a television signal;

Figure 4 is a block diagram illustrating the details of a transaction enabler in an embodiment of the present invention;

Figure 5 depicts a display screen using which a user may participate in auctions in accordance with the present invention.

Detailed Description of the Preferred Embodiments

1. Overview and Discussion of the Invention

The present invention allows viewers ("television viewers") of television systems to participate in auctions. Typically, the data relating to an item ("auction item") offered for sale in an on-going auction is encoded in a television signal. The encoded information may be displayed while the television viewers watch the images encoded in the television signal. The

viewers may be provided a convenient interface to bid on the auction item.

Auction items consistent with expected viewer profiles may be sold using the present invention. A seller may be able to sell at higher prices as many viewers are likely to bid. For example, a diamond ring may be auctioned towards the end of a romantic movie. The invention is described below with respect to several examples for illustration.

2. Example Environment

Figure 1 is a block diagram illustrating an example environment in which the present invention can be implemented. The environment may include bidding systems 110-A and 110-B, Internet 120, web site 130, broadcast system 150, and television 170. A viewer of television 170 may participate in auctions as described below in further detail.

Web site 130 may provide an auction service. As an illustration, web site 130 may implement the interface of www.ebay.com, well known in the relevant arts. Bidder systems 110-A and 110-B may access Internet 120 to bid on the items offered for sale on web site 130. Bidding systems 110-A and 110-B, Internet 120, and web site 130 may be implemented in a well-known way. Even though the auction service is shown as being provided from web site 130, it should be understood that different other servers using different access technologies (e.g., dial-up) may be used in providing the service.

Broadcast system 150 includes information related to an auction item in a television signal and transmits the television signal on broadcast medium 146 (airwaves, cable, etc.). The data may specify the item offered for sale, the present highest bid, and an access address for enabling the viewer to bid. For example, the access may contain a URL of web site 130. An example embodiment of broadcast system 150 is described below.

The auction may be in progress (on-going) on web site 130, and accordingly broadcast system 150 may access web site 130 to access any data (e.g., present highest bid) for inclusion

in the television signal. Link 134 may be provided on Internet 120 even though a dedicated line is shown in Figure 1.

Viewer bidding system 150 receives the television signal, and enables a viewer to participate in auctions. Viewer bidding system 150 may display the images encoded in the received television signal. In addition, viewer bidding system 150 may recover the data
5 related to the auction item, and display the corresponding information. By appropriate action, the user may indicate a higher bid and transmit the higher bid on virtual link 163 on Internet 120.

In an embodiment, viewer bidding system 150 may include television 170, transaction
10 enabler 160, and remote control 180. Transaction enabler 160 may overlay any images necessary for providing an user interface on top of the images encoded in the television signal (“television signal images”). For example, information identifying the auction item (e.g. Mark McGuire’s bat) and the highest bid price may be overlaid on television signal images.

Transaction enabler 160 may encode the overlaid image in a form consistent with
15 conventional television signals for display on television 170. In other words, transaction enabler 160 operates as a ‘set-top’ box. However, transaction enabler 160 may be integrated into television 170, for example, using embedded chip-sets provided by TeleCruz Technology, Inc. In either case, remote control 180 enables the user to specify the bid price and to transmit the new bid. An example embodiment of transaction enabler 160 is described below in further
20 detail. However, first a method in accordance with the present invention is described first below.

3. Method

Figure 2 is a flow-chart illustrating a method in accordance with the present invention. The method begins in step 201, in which control passes to step 210. In step 210, data identifying an auction item and an access address may be encoded in a television signal. The data identifying an auction item may include both a descriptive component (e.g., "baseball bat signed by Mark McGuire") and a unique code specifying the auction item (or group in case multiple items of the same type are available).

The data may be encoded in one of different formats depending on different criteria, but consistent with an interface at viewer bidding system 150. For example, a unique code identifying an auction item may be encoded as a parameter of a URL (access address) since the web browser's based technology lends well to such encoding and later submission of a bid. The television signal may also be encoded with image frames for display on television signals. Both (images and data related to auction items) encoding may be performed in a known way.

In step 220, the television signal may be broadcasted to television systems covering a large geographic area. In step 240, the television signal may be received at a viewer end (e.g., by transaction enabler 160 of Figure 1). In step 260, the data related to the auction item (encoded in step 210) may be recovered. The recovery generally needs to be consistent with the encoding scheme used by broadcast system. In general, any compatible encoding scheme may be used.

In step 280, the user is provided a convenient user interface to bid on the auction item. Typically, the description of the auction item is displayed, and the user may be provided the option to bid, in which case the bid is submitted to a system identified by the access address. While submitting the bid, the unique code identifying the auction item may be used to specify to the system that the bid relates to that particular item. The access address is used to connect

the user to a central machine (e.g., web site or any server) or person. The user may then submit the bid. The highest bidder is generally entitled to the offered auction item for the submitted bid.

The method and environment described above may be applied in several ways as will be apparent to one skilled in the relevant arts based on the disclosure herein. All such implementations are contemplated to be within the scope and spirit of the present invention. However, it may be desirable to have bidders (television viewers) participation at different points of a broadcast. The manner in which the point can be controlled is described below with respect to broadcast system 140.

10 4. Broadcast System

Figure 3 is a block diagram illustrating an example embodiment of broadcast system 140. Even though the description of broadcast system is provided substantially with respect to broadcasters producing a television signal, the present invention can be practiced by intermediate broadcasters also. Such advertisements are generally more targeted to the specific geographic profile. Broadcast system 140 may contain production block 310, authoring block 320, broadcast block 330, timing determination block 340, auction data interface 360, and storage 350. Each block is described in further detail below.

Timing determination block 340 may determine the specific time at which to encode data related to an auction item. For example, it may be desirable to broadcast data related to a baseball bat (auction item) when a home run is hit. Timing determination block 340 may be implemented to monitor the scores of the baseball game and generate an indication to auction data interface 360. Several other criteria can be used in determining when to send data related to an auction item.

Timing determination block 340 may also determine when to send updates

corresponding to various auctions. When timing determination block 340 determines to cause update corresponding to an auction to be sent, auction data interface 360 may interact with web site 130 to retrieve a present highest bid from web site 130. The present highest bid may be provided to authoring block 320 for encoding in a broadcast television signal.

5 Auction data interface 360 receives data on line 134 if a web based auction is on-going for the auction item of interest on web site 130. The data may indicate the present highest bid, bid history, the seller, any comments about the seller. As noted above, auction data interface 360 may provide the data to be encoded in the television signals. The data may contain, in addition to the data retrieved from web site 130, data identifying the auction item (descriptive
10 component and unique code).

Some of the data may be pre-stored in storage 350 also. For example, it may be desirable to display graphic icons on television systems to represent different auction items. Bit maps representing the graphics icons may be stored in storage 350. In general, auction data interface 360 may gather any data which may be of interest to bidders, and pass the data
15 to authoring block 320.

Production block 310 may contain different components such as cameras which are used to film a show/program. The display signal is preferably in a form suitable for eventual transmission as a television signal. In general, production block 310, may encode images in a display data portion of a television signal. The images may be displayed later on a television
20 system for viewing a broadcast program. Production block 310 may be implemented in a known way.

Authoring block 320 encodes data received from auction data interface 360 into television signals. The data may be encoded according to any convention, and transaction enabler 160 may need to be accordingly designed. Several such conventions can be designed

in known way. Authoring block may either store the resulting signal in storage 350 or forward to broadcasting block 330.

In one embodiment, authoring block 320 encodes the data in non-display portion (e.g., vertical blanking interval) of the display signal. Such encoding may be performed in a known way. In an alternative embodiment, the data may be encoded in other portions (e.g., least significant bits of pixel data elements representing an image) as well. This alternative embodiment is described in further detail in co-pending U.S. Patent Application Entitled, "Encoding Hot Spots in Television Signals", Serial Number: 09/276,266, Filing Date: March 25, 1999, which is incorporated in its entirety into the present application.

Even though the encoding is described with reference to analog television signals, it should be understood that the present invention may be practiced in conjunction with digital television signals (e.g., those suitable for HDTV) also. Some of the techniques described in this application may be employed for such encoding in the digital television signals. Many other techniques will be apparent to one skilled in the relevant arts based on the disclosure herein. Such other techniques are also contemplated to be within the scope and spirit of the present invention.

Broadcast block 330 may broadcast television signals (containing the hot spot data in the display data portion) in a known way. It should be noted that the television signal can be in progressive scan format or interlaced format. Production block 310 and authoring block 320 need to be implemented taking into consideration the transmission standard (progressive vs. interlaced, and digital vs. analog) of the television signals. Thus, broadcast block 330 generates television signals containing data which may be used to enable television viewers to bid on the auction items.

Transaction enabler 160 receives the television signals and enables a viewer to bid on

the auction items. Example embodiments of transaction enabler 160 are described below in further detail. Before describing example embodiments of transaction enabler 160 in detail, it is helpful to understand some typical problems with the user interface.

5. Problems and Solutions

5 In one embodiment, a highest present bid may be encoded in the television signal, and the user may submit a higher bid than the highest present bid. One problem associated in the environments of Figures 1 and 2 is that many bidders may bid for the auction item based on the same highest bid. As the bids are generally marginally more than the present highest bid, the approach may not maximize the return for the seller.

10 Accordingly, an improvement may be implemented in which an "auction close time" (time at which the auction for the auction item ends) may be associated with the auction item. The auction close time may also be encoded and transmitted in the television signals. Thus, viewers may choose a later convenient time for bidding on the auction item. However, in such a situation, viewer bidding system 150 may need to store the required data.

15 Yet another problem is, a viewer may wish to know an updated highest bidding price before actually submitting a bid. Thus, the viewer may be provided a convenient user interface to request a 'present highest bid' associated with an auction item of interest. The updated price may also be received on virtual link 163. In this case also, viewer bidding system 150 may need to store the required data.

20 In yet another scenario, a viewer may wish continuous updates of the highest bidding price. Accordingly, a viewer may be provided an option of initiating a small window in which the updates to the highest bids are provided continuously (e.g., when highest bid changes or every 3 seconds). An embodiment of transaction enabler 160, which provides for at least these features is described below.

6. Transaction Enabler

Figure 4 is a block diagram illustrating the internals of an example embodiment of transaction enabler 160 containing image decoder 410, memory 430, recovery block 420, processor 450, digital to analog converter (DAC) 485, multiplexor 480, infra-red (IR) receiver 460, telephone interface 470 and broadband interface 475. Each component is described
5 below in further detail.

Image decoder 410 generates pixel data elements representing image frames encoded in a television signal received on broadcast channel 146. In response to the operation of remote control unit 180, image decoder 410 may store the pixel data elements representing an
10 image frame in memory 430. Such storage enables overlays. Image decoder 410 may be implemented in a known way. Memory 430 may represent several memory modules such as fast random access memories and relatively slower non-volatile memories. The non-volatile memories may store data and program instructions which enable the operation of the present invention.

15 Recovery block 420 recovers the data related to auction items encoded in the received television signal. In general, recovery block 420 needs to be implemented consistent with any conventions or protocols used at broadcaster end 380 for encoding the hot spot data. If the data is encoded in non-display portions (e.g., VBI), the data may be recovered in a known way. If the data is encoded in display data portion (i.e., in images), recovery block 420 may
20 examine the pixel data elements stored in memory 430 to recover the data. Further details of recovery are noted in co-pending U.S. Patent Application Entitled, "Encoding Hot Spots in Television Signals", Serial Number: 09/276,266, Filing Date: March 25, 1999, which is incorporated in its entirety into the present application.

Infra-red (IR) receiver 460 receives remote control signals from remote control unit 180, and provides digital data representing the remote control signals to processor 450. The control signals may indicate whether the user wishes to see auction item related data, to enter the bid, to receive an updated present highest bid, etc. Several features of the user interface
5 may be activated by a viewer using IR receiver 460. IR receiver 460 may be implemented in a known way. It may be noted that other receivers which receive control signals from viewers and provide corresponding digital data to processor 450 may be implemented.

Telephone interface 470 enables a telephone call to be initiated. Such telephone calls may be generally initiated either to connect to the Internet via an ISP or to contact a phone
10 with a live-operator. When a telephone call is initiated with a live operation, telephone interface 470 may provide the necessary micro-phone (for a viewer to speak) and receiver for reproducing audible voice. Alternatively, a user may utilize a conventional telephone set that is attached to line 335.

Broadband interface 475 may provide a high speed connection (e.g., using a local area
15 network, digital subscriber loop technology or cable interface) to connect with a web server (corresponding to an URL) or even initiate a voice call (e.g., using voice over Internet Protocol). Telephone interface and broadband interface may be logically viewed as being part of line 163 of Figure 1. In general, broadband interface 475 and telephone interface 470 provide the communication to a system (specified by access address) providing auction
20 service.

Processor 450 receives data related to auction items from recovery block 420, and enables a user to send a bid to a system identified by an access address. The transmission of the bid may be either by broadband interface 475 or telephone interface 470 as specified by the type of access address. Processor 450 may also implement the user interface features noted

in the section above.

For a suitable user-interface, processor 450 may control the images displayed on television system 110. For example, processor 450 may overlay information in the auction items related data on the television signal image. Specifically, the portion to be overlaid on television images may be provided by processor 450, and control line 481 may be controlled to accomplish the overlay function. However, when a user does not wish to bid or when data related to auction items is absent in television signals, processor 450 may control select line 481 to cause the television signal received on line 146 to be passed directly on line 167. In addition, processor 450 may cause auction related data to be displayed in a transparent mode. Typically, techniques such a half-tone control are used for achieving such transparency of display.

If the access address is a URL, transaction enabler 160 may need to operate as a web-browser. Processor 450 may enable such an operation by executing the program instructions provided by memory 430. The web-browser enables transaction enabler 160 to receive different web-pages in a known way. Processor 450 may convert the web pages into image frames, and encode the image frames into a television signal having a format compatible with conventional television signals such that the images can be displayed on television system 110. Well known methods may be employed for such conversion and encoding.

Therefore, transaction enabler 160 may operate in conjunction with broadcast system 140 to enable a television viewer to participate in auctions. As a result, viewers of television systems may be drawn to participate in auctions which are generally accessed mostly by users surfing the world-wide-web. It should be understood that web site 130 and broadcast system 140 may be integrated as one unit depending on the available technologies, and in such a case,

transaction enabler 160 may communicate with such a unit directly. The present invention is described in further detail below with reference to an example user interface considering some of the description of above.

7. User Interface

5 Figure 5 is a diagram illustrating the manner in which transaction enabler 160 may enable a viewer of television programs to participate in auctions. It should be understood that transaction enabler may use other display devices from which a user can participate in auctions. In addition, other types of systems (such as computers) which display images in television signals may also be used to participate in auctions in accordance with the present
10 invention.

Continuing the description with reference to Figure 5, there is shown television display 500 (for example, on television 170). Auction related data may be received in accordance with the present invention, and the relevant data may be displayed in a small window 540. Window 540 is preferably overlaid on television program images as a transparent window
15 using techniques such as half-toning well known in the relevant arts. By using a transparent display, a viewer may be able to watch the programs encoded in the television signal while participating in the auctions.

Window 540 may be used to display the description of the auction item ("McGuire's
70th Home Run Bat" in the example there), the present highest bid, bidder of the highest bid,
20 and the time at which the auction for this item is expected to close may be displayed. The present highest bid may be periodically updated using the data received on the broadcast television signal. On the other hand, a viewer may select (click on) 'Update' text to cause transaction enabler 160 to initiate a dialogue with web server 130, and retrieve updated information for a presently watched auction item. Thus, in Figure 5, such a selection may

cause transaction enabler 160 to display \$4300 (representing an increase in the present highest bid).

The user may select 'Bid History' to view the previous bidders and history. The relevant data may either be displayed based on data stored locally or the data may be retrieved
5 from web site 130 in response to a user request. As is well known in the relevant arts, auction sites such as www.ebay.com provide such bid histories.

The user may specify her/his bid price in the box provided next to text 'Your Bid'. The user may then select the 'Submit' text to cause transaction enabler 160 to submit the bid. As noted above, the submission may be according to any mechanism. The bid can potentially
10 be over a broadband interface to access a web site or to a server accepting over a telephone connection. Once the bid is submitted to a server at the access address, the auction item may be sold to a bidder in a known way. If the user of system 150 has the highest bid, the user may pay the bid amount and receive the auction item.

Thus, an interface such as the one above, a user (or television viewers) may bid for
15 auction items in accordance with the present invention. The bid may be submitted according to any pre-specified protocol between transaction enabler 160 and an auction server (e.g., web site 130). The implementation of auction on web site 130 based on such received bid prices will be apparent to one skilled in the relevant arts.

8. Conclusion

20 While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What Is Claimed Is:

- 1 1. A method of enabling a viewer of a television system to participate in auctions, said
2 method comprising:
- 3 (a) encoding in a television signal a data describing an auction item and an access
4 address of a server at which auction service for said auction item is provided; and
5 (b) transmitting said television signal,
6 wherein said data can be used to enable said viewer to bid for said auction item at said
7 server.
- 1 2. The method of claim 1, wherein said method further comprises:
- 2 (c) receiving said television signal encoded with said data in a transaction enabler;
3 (d) recovering said data encoded in said television signal;
4 (e) displaying information describing said auction item on said television system;
5 (f) enabling said viewer to bid at said server specified by said access address.
- 1 3. The method of claim 2, further comprising:
2 (g) enabling said viewer to specify a bid price for said auction item.
- 1 4. The method of claim 3, wherein said enabling said viewer to specify said bid price
2 comprises:
3 (h) enabling said viewer to indicate said bid price; and
4 (i) transmitting said bid price to said server at said access address.
- 1 5. The method of claim 4, wherein said access address comprises a telephone number

2 of said server, and said method further comprises:

3 (j) encoding a unique code identifying said auction item;

4 (k) recovering said unique code in said transaction enabler; and

5 (l) transmitting said unique code along with said bid price to said server,

6 whereby said server can easily associate said bid price with said auction item using said

7 unique code.

1 6. The method of claim 4, wherein said access address comprises a universal resource
2 locator (URL) of a web site, wherein said web site comprises said server, and wherein steps
3 (h) and (i) comprise the further step of enabling said viewer to indicate said price on a web
4 page provided by said web site.

1 7. The method of claim 1, further comprising:

2 (m) encoding a present highest bid in said television signal, wherein said present
3 highest bid may be displayed to said viewer before said viewer decides to submit a bid.

1 8. The method of claim 7, wherein said server comprises a web site, and said method
2 comprising the further step of retrieving said present highest bid from said web site.

1 9. The method of claim 1, wherein step (a) comprises the step of encoding said data
2 in non-display portion of said television signal.

1 10. The method of claim 1, wherein step (a) comprises the further step of encoding
2 said data in a non-display portion of said television signal.

1 11. The method of claim 10, wherein said non-display portion comprises vertical
2 blanking interval (VBI).

1 12. The method of claim 1, further comprising:
2 transmitting an updated highest bid price in said television signal, wherein said updated
3 highest bid price corresponds to a present highest bid for said auction item.

1 13. The method of claim 12, further comprising:
2 retrieving said updated bid price from said server,
3 wherein said step of transmitting said updated highest bid price is performed after said
4 step of retrieving said updated bid price from said server.

1 14. The method of claim 13, further comprising:
2 enabling said viewer to request a bid history; and
3 displaying all of said updated bid prices to said viewer.

1 15. The method of claim 14, wherein said display corresponding to said bid history
2 further comprises a description of the bidder corresponding to each of said present highest bid.

1 16. The method of claim 1, wherein said data further comprises a time at which
2 auction for said auction item closes.

1 17. A method of enabling a viewer of a television system to participate in auctions,

2 said method comprising:

- 3 (a) receiving in a transaction enabler a television signal encoded with a data, said data
4 including a description of an auction item and an access address of a server at which auction
5 service for said auction item is provided;
- 6 (b) recovering said data encoded in said television signal;
- 7 (c) displaying said description of said auction item on said television system;
- 8 (d) enabling said viewer to bid at said server specified by said access address.

1 18. The method of claim 17, further comprising:

- 2 (e) enabling said viewer to indicate said bid price; and
- 3 (f) transmitting said bid price to said server at said access address.

1 19. The method of claim 4, wherein said access address comprises a telephone number
2 of said server, and said method further comprises:

- 3 (g) encoding a unique code identifying said auction item;
- 4 (h) recovering said unique code in said transaction enabler; and
- 5 (i) transmitting said unique code along with said bid price to said server,
6 whereby said server can easily associate said bid price with said auction item using said
7 said unique code.

1 20. An environment enabling a viewer of a television system to participate in auctions,
2 said environment comprising:

3 encoding means for encoding in a television signal a data describing an auction item

4 and an access address of a server at which auction service for said auction item is provided;
5 and
6 transmission means for transmitting said television signal,
7 wherein said data can be used to enable said viewer to bid for said auction item at said
8 server.

1 21. An environment enabling a viewer of a television system to participate in auctions,
2 said environment comprising:

3 receiving means for receiving a television signal encoded with a data, said data
4 including a description of an auction item and an access address of a server at which auction
5 service for said auction item is provided;

6 recovery means for recovering said data encoded in said television signal;

7 displaying means for displaying said description of said auction item on said television
8 system;

9 enabling means for enabling said viewer to bid at said server specified by said access
10 address.

1 22. An environment enabling a viewer of a television system to participate in auctions,
2 said environment comprising:

3 a broadcast system to encode in a television signal a data describing an auction item
4 and an access address of a server at which auction service for said auction item is provided,
5 said broadcast system being designed also to transmit said television signal,

6 wherein said data can be used to enable said viewer to bid for said auction item at said
7 server.

1 23. The environment of claim 22, wherein said broadcast system comprises:
2 a production block to generate images to encode in a display data portion of said
3 television signal;
4 an authoring block to encode said data in said television signal; and
5 a broadcast block to transmit said television signal containing said images and said
6 data.

1 24. The environment of claim 23, further comprising an auction data interface to
2 receive a present highest bid from a server, said auction data interface to provide said present
3 highest bid to said authoring block, wherein said authoring block encodes said present highest
4 bid in said television signal.

1 25. The environment of claim 24, further comprising a timing determination block to
2 determine the time at which said authoring block encodes said data including said present
3 highest bid in said television signal.

1 26. The environment of claim 22, further comprising:
2 a viewer bidding system to receive said television signal, and enabling said viewer to
3 submit a bid and participate in said auction.

1 27. The environment of claim 26, wherein said viewer bidding system comprises:
2 a television system;
3 a remote control which enables said viewer to submit said bid; and

4 a transaction enabler coupled to said television system and to receive said commands
5 from said remote control, said transaction enabler to recover said data encoded in said
6 television signal and display information contained in said data on said television,
7 wherein said viewer can submit said bid using said remote control.

1 28. The environment of claim 27, wherein said transaction enabler is integrated within
2 said television system.

1 29. The environment of claim 27, wherein said transaction enabler is provided external
2 to said television system, and wherein said transaction enabler overlays a window with
3 information contained in said data on images encoded in the display data of said television
4 signal.

1 30. The environment of claim 27, wherein said window is displayed in a transparent
2 mode on said images.

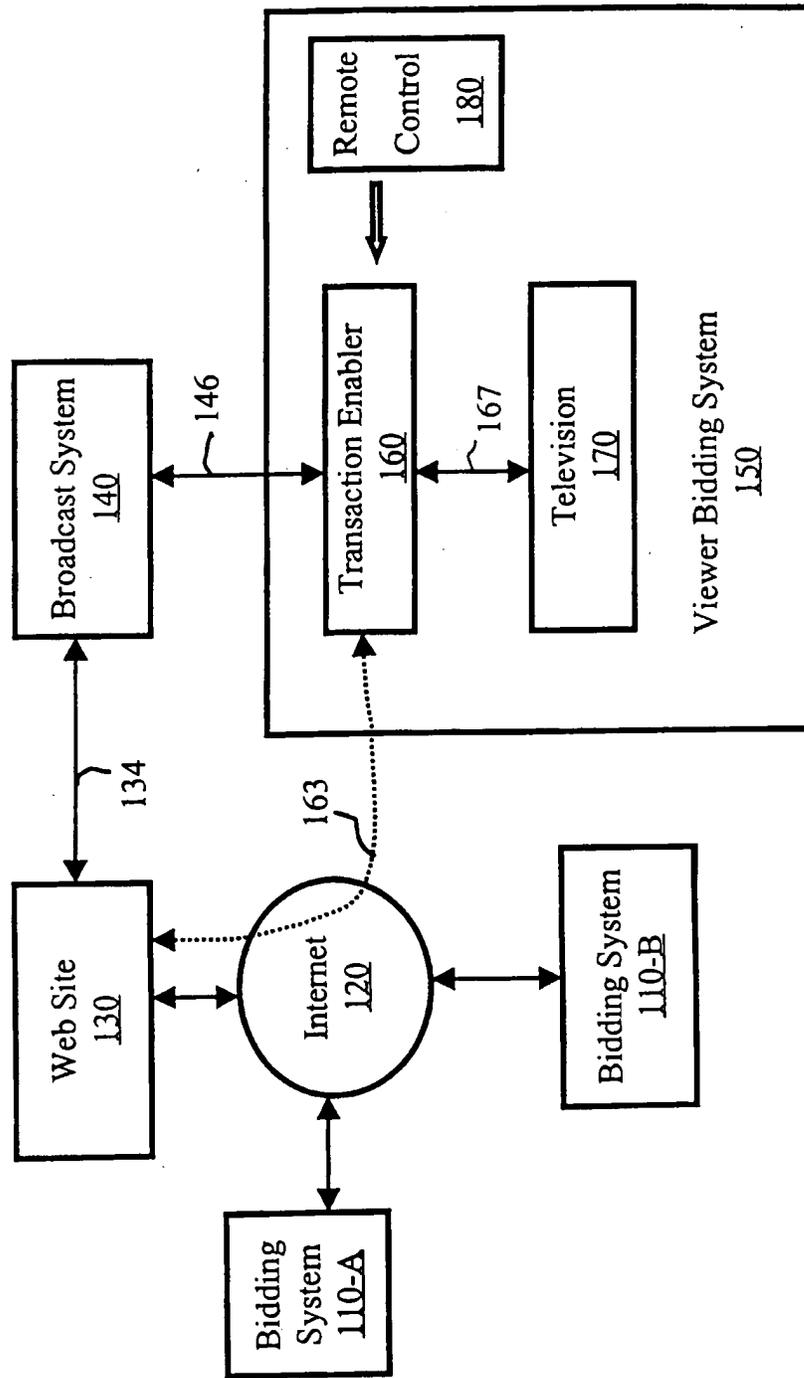


Figure 1

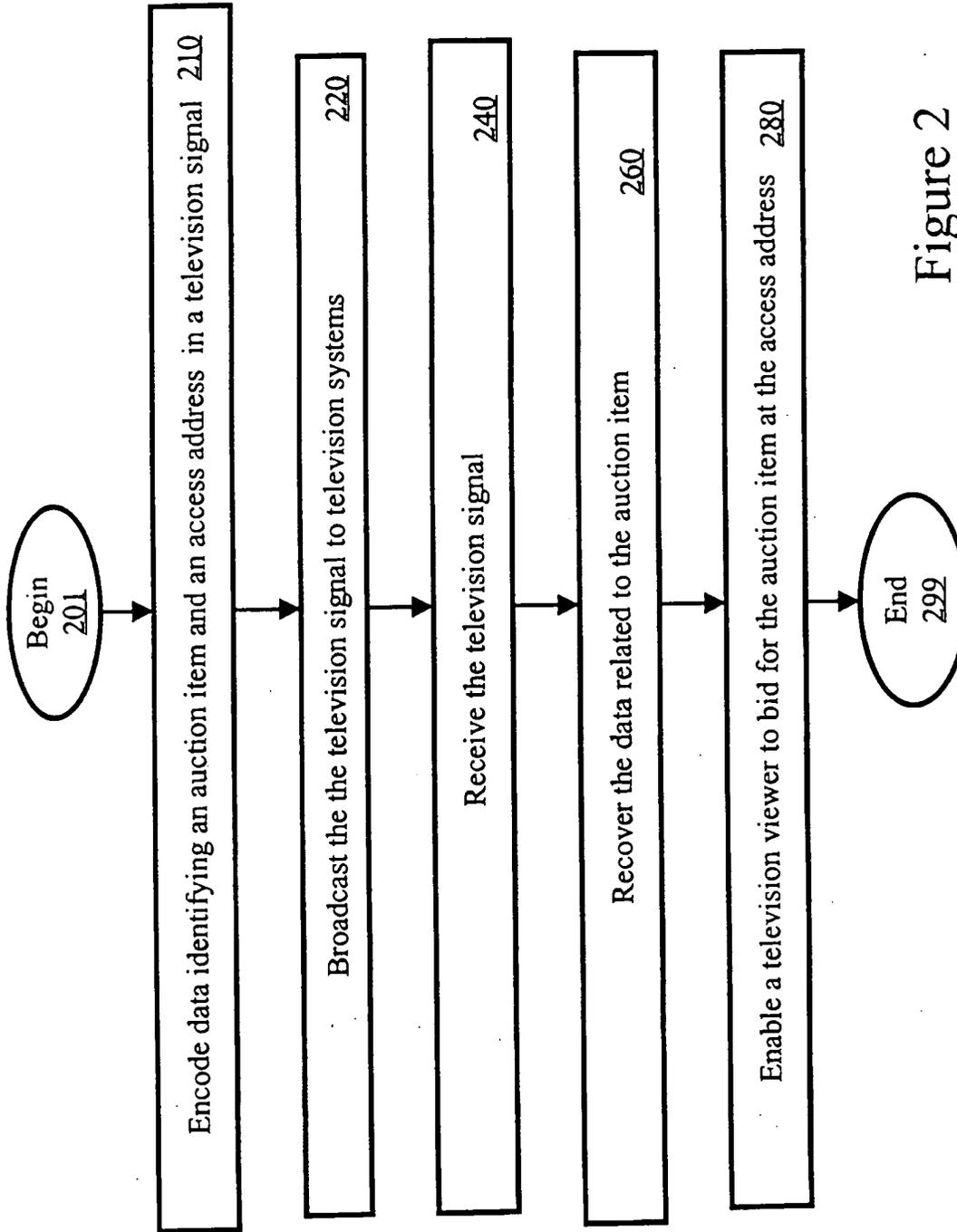


Figure 2

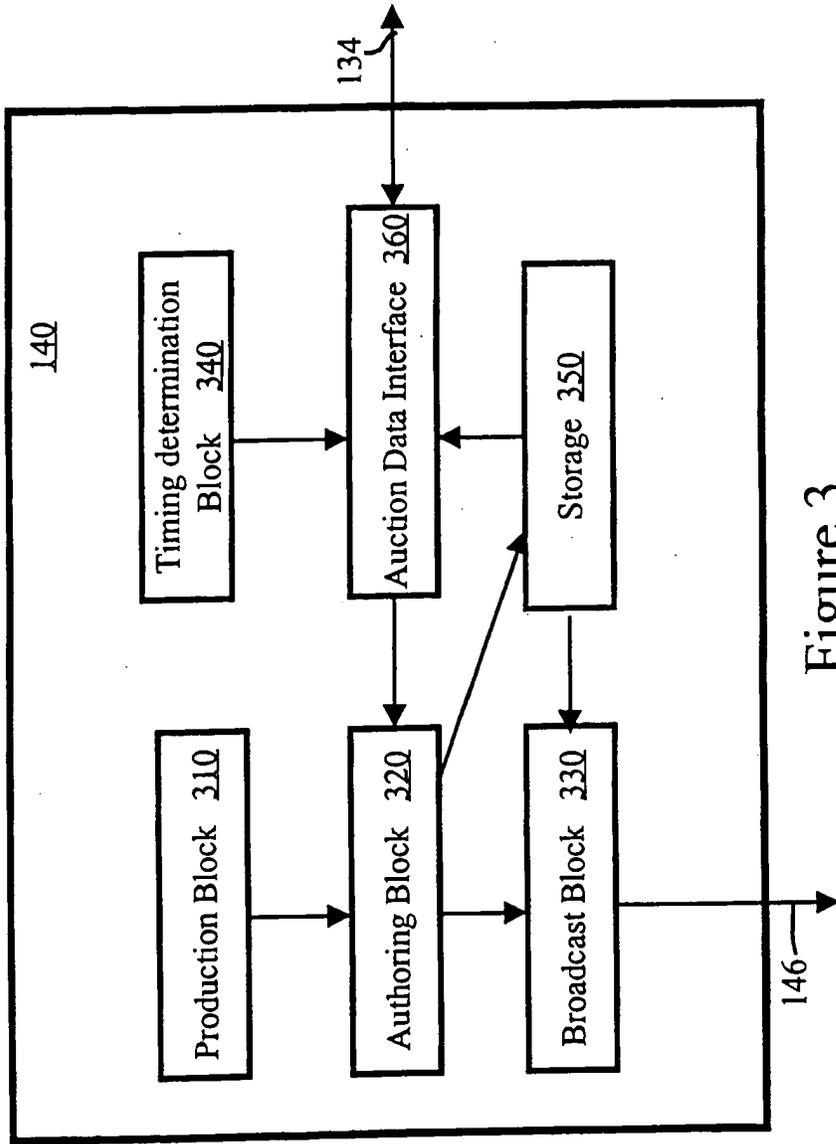


Figure 3

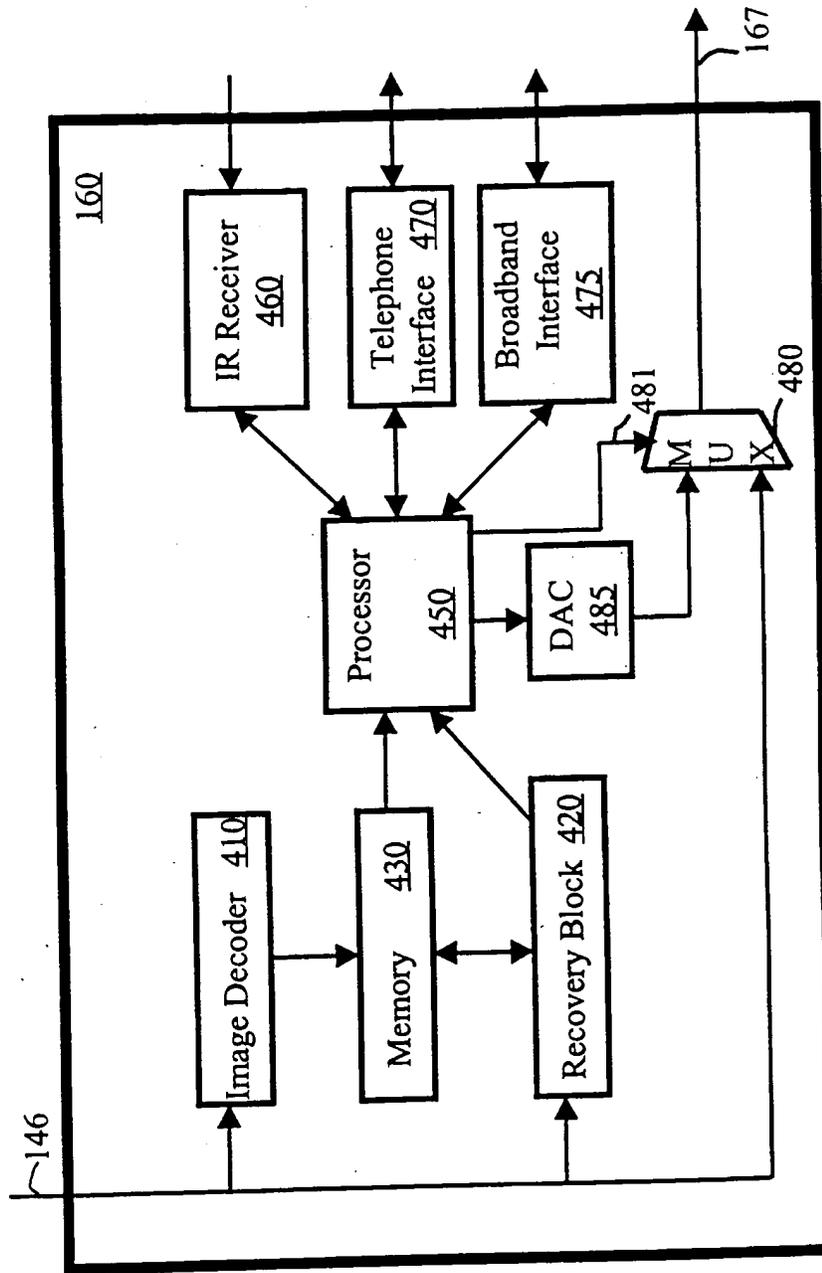


Figure 4

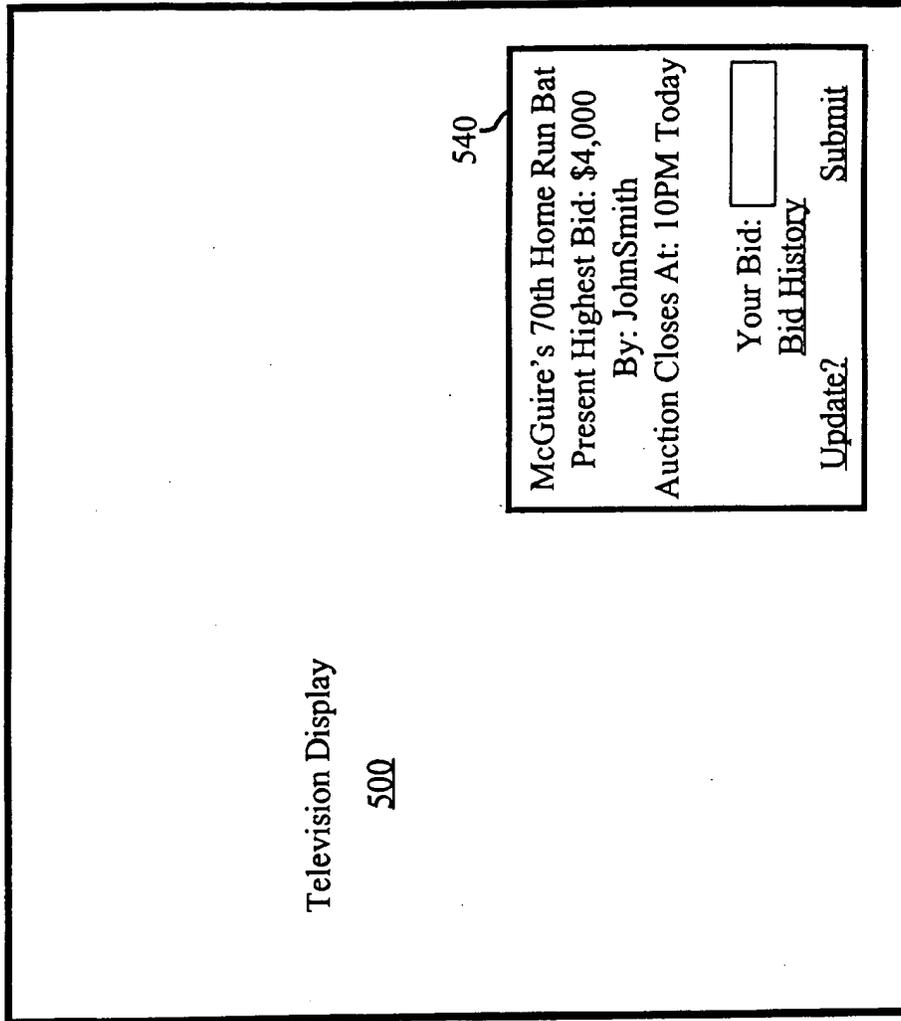


Figure 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/18510

A. CLASSIFICATION OF SUBJECT MATTER IPC(7) :G06F 17/60 US CL : 705/26, 27, 37 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) U.S. : 705/26, 27, 37 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Please See Extra Sheet. Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EAST, CORPORATE RESOURCE NET		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	Auction Goes Upscale. Capital District Business Review. April 17, 1995. Vol. 22. Issue 1. page 43.	1-30
Y,E	Strategic Partnership Between ExtraLot.com and The Auction Channel. Business Wire. August 11, 2000.	1-30
Y	Auctioneer Onsale to Broadcast Live Commercials on ZDTV. Electronic Advertising and Marketplace Report. October 6, 1998. Vol 12. Issue 18. page 4.	1-30
Y	Philadelphia Business Journal. Auction Television Does \$1 Million Stock Placement. January 29, 1999. Vol. 17. Issue 51. page 36.	1-30
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents:		*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance		*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date		*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)		*G* document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means		
P document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search 22 AUGUST 2000	Date of mailing of the international search report 18 SEP 2000	
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer JAMES TRAMMELL <i>Rugenia Zagan</i> Telephone No. (703) 305-1760	

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/18510

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,905,975 A (AUSUBEL) 18 May 1999, col 3, lines 1-30.	1-30
Y	MARQUEZ, RACHELLE. New Dimension For Auction. 15 September 1997. Vol. 15. Issue 20. page 38.	1-30

Form PCT/ISA/210 (continuation of second sheet) (July 1998)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/18510

B. FIELDS SEARCHED

Documentation other than minimum documentation that are included in the fields searched:

NEWTON'S TELECOM DICTIONARY
McGRAW-HILL ENCYCLOPEDIA OF ELECTRONICS AND COMPUTERS



(WO/2004/103843) PACKAGING METHOD AND DEVICE, PACKAGING BAGS

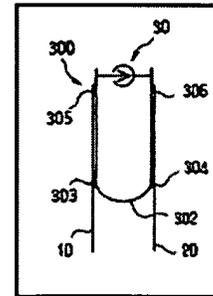
Biblio. Data	Description	Claims	National Phase	Notices	Documents
--------------	-------------	--------	----------------	---------	-----------

Latest bibliographic data on file with the International Bureau

Publication Number: WO/2004/103843 International Application No.: PCT/FR2004/001185
 Publication Date: 02.12.2004 International Filing Date: 14.05.2004
 Int. Class.: B65D 33/25 (2006.01), B65D 85/16 (2006.01)
 Applicants: S2F FLEXICO [FR/FR]; 1, route de Méru, F-60119 Henonville (FR) (All Except US).
 BOIS, Henri, Georges [FR/FR]; 61, boulevard d'Inkermann, F-92200 Neuilly sur Seine (FR) (US Only).
 Inventor: BOIS, Henri, Georges [FR/FR]; 61, boulevard d'Inkermann, F-92200 Neuilly sur Seine (FR).
 Agent: MARTIN, Jean-Jacques; Cabinet Regimbeau, 20, rue de Chazelles, F-75847 Paris Cedex 17 (FR).
 Priority Data: 03/05887 16.05.2003 FR

Title: PACKAGING METHOD AND DEVICE, PACKAGING BAGS

Abstract: The invention relates to a packaging method comprising the following steps: provision of a bag whose mouth comprises opening/closing means (30) for multiple successive openings and closings and a cleavable linking veil, located at a distance therefrom inside the bag in relation to said opening/closing means (30); introduction of contents (100) to be wrapped in the bag and tightening of said bag in order to close it, tension being applied to the contents (10); the veil (40) enters into contact with the contents (100) avoiding the application of stress on the opening/closing means, guaranteeing free access to the contents (100) via said opening/closing means (30) after tearing, enabling the bag to be relaxed in a closed state as a result of the distance (D) separating the veil (40) and the opening/closing means (30). The invention also relates to a packaging device and to bags thus obtained.



Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
 African Regional Intellectual Property Org. (ARIPO) (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW)
 Eurasian Patent Organization (EAPO) (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM)
 European Patent Office (EPO) (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR)
 African Intellectual Property Organization (OAPI) (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Publication Language: French (FR)
 Filing Language: French (FR)

(19) World Intellectual Property Organization International Bureau



(43) International Publication Date 22 April 2004 (22.04.2004)

PCT

(10) International Publication Number WO 2004/034223 A2

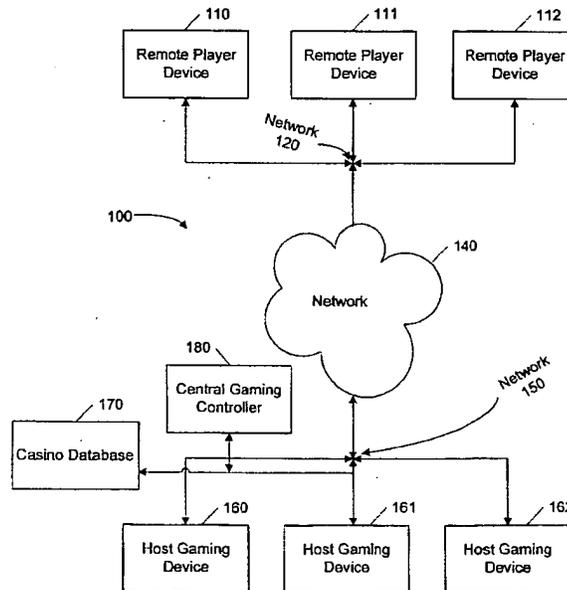
- (51) International Patent Classification⁷: G06F
- (21) International Application Number: PCT/US2003/032153
- (22) International Filing Date: 8 October 2003 (08.10.2003)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 60/417,913 9 October 2002 (09.10.2002) US
- (71) Applicant (for all designated States except US): LEGAL IGAMING, INC. [US/US]; 200 Ultra Drive, Henderson, NV 89074 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): SAUNDERS, Michael, W. [US/US]; 200 Ultra Drive, Henderson, NV 89074 (US). MILLER, William, D., III [US/US]; 8740 Country Pines Avenue, Las Vegas, NV 89129 (US). CARLSON, Rolf, E. [US/US]; 211 Dartmouth Avenue S.E., Albuquerque, NM 87106 (US).

- (74) Agent: MALLON, Joseph, J.; Knobbe, Martens, Olson & Bear, LLP, 2040 Main Street, 14th Floor, Irvine, CA 92614 (US).
- (81) Designated States (national): AE, AG, AL, AM, AT (utility model), AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (utility model), CZ, DE (utility model), DE, DK (utility model), DK, DM, DZ, EC, EE (utility model), EE, EG, ES, FI (utility model), FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK (utility model), SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17: of inventorship (Rule 4.17(iv)) for US only

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR CONNECTING GAMING DEVICES TO A NETWORK FOR REMOTE PLAY



(57) Abstract: A system (100) and method for connecting remote player devices (110) to regulated host gaming devices (160) in a network to provide remote game play. A host gaming device (160) is configured to provide game information to a plurality of remote player devices (110) to allow remote play of the host game device (160). Whether each remote player device (110) is permitted to receive gaming data is based upon, at least in part, the geographic location of the remote player device (110).

WO 2004/034223 A2



Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD FOR CONNECTING GAMING DEVICES TO A NETWORK FOR REMOTE PLAY

Background of the Invention

Field of the Invention

[0001] The present invention generally relates to electronic devices. In particular, the invention relates to methods and systems of interactive gaming.

Description of the Related Technology

[0002] Traditionally, the way for a gaming operator to increase revenue from gaming devices is to increase the number of gaming devices available for play. In order for casinos to increase the number of gaming devices available for play, casino floor space must be added to house the additional gaming devices. The floor space allocated to house additional gaming devices must meet specific criteria as defined by the gaming authority for the jurisdiction in which the gaming devices are to be located. Providing additional floor space is an expensive process for casino operators and often requires constructing new casino properties. Also, adding gaming devices typically requires payment of additional licensing fees for each additional game.

[0003] A trend in the gaming industry has been to provide Internet gaming. Internet gaming allows players to make wagers on the outcome of casino style games similar to that described above, except that the player does not have to be physically located in a casino to do so. Internet players make wagers and play casino games using a personal computer and wager on games running on computers connected to the Internet.

[0004] More broadly, interactive gaming is the conduct of gambling games through the use of electronic devices. The popularity of Internet gambling sites has indicated a strong market for remotely accessible gaming, or other interactive gaming. Regulated casino operators strongly desire to provide interactive gaming while capitalizing on existing infrastructure. Thus there is a need for improved electronic devices that support regulated remote gaming.

Summary of the Invention

[0005] The system of the present invention has several aspects, no single one of which is solely responsible for its desirable attributes. Without limiting the scope of this invention as expressed by the claims which follow, its more prominent features will now be discussed briefly. After considering this discussion, and particularly after reading the section entitled "Detailed Description of the Invention" one will understand how the features of this invention provide advantages which include providing remote gaming in regulated environment.

[0006] A gaming system and method of using the same to allow a host gaming device to be played from remote player devices to allow casino operators to obtain maximum advantage from their gaming licenses.

[0007] More particularly, in one embodiment gaming system may comprise a data network, a host gaming device connected to the data network, the gaming device configured to execute at least one game and a plurality of remote player devices connected to the data network. Each of the remote player devices is configured to receive game information provided by the host gaming device. Whether each remote player device is permitted to receive gaming data may be based upon, at least in part, the geographic location of the remote player device.

[0008] The host gaming device may be configured to allow no more than a predetermined number of remote player devices to concurrently receive game information provided by the host gaming device during the gaming session. This predetermined number may be determined by a gaming agency.

[0009] In another embodiment of a gaming system, at least one of the plurality of remote player devices may be permitted to receive game data based upon, at least in part, the geographic location of the remote player device, an age of a user of the remote player device.

[0010] A gaming system according to the invention may also include a central gaming controller configured to record gaming transactions on the host gaming device and on each remote gaming device.

[0011] The data network may be, in part, the Internet, and be comprised of one or more logical segment, which may include closed-loop networks. The host gaming device may be configured to identify the geographic location of a remote player device based, at least in part, on a logical segment corresponding to the remote player device. A mobile communications network, or a GPS device may also allow identification of the geographic location of the remote player device.

[0012] The host gaming device may be in a location approved by a gaming agency and include at least one game control configured to provide local use. This game control may be disabled when the host gaming device is providing game information to a remote player device. A host gaming device may also be configured to save an encrypted game state allowing a game to be resumed following a device or network failure.

[0013] A remote player device may be coupled to a credential device configured to receive information relating to a user of the remote player device. The information relating to a user may include the age of the user, or a password that is input by the user. The credential device is a smart card reader, a biometric device such as a fingerprint reader, or any type of input device. The credentials may be verified against information, such as age, password, or fingerprint in a database configured to provide information associated with each of a plurality of users of the gaming system.

[0014] In another embodiment, a gaming system may be comprised of a means for executing at least one game, the game providing game information during its execution, a local access means provides local access to the game information for a user in a location approved by a gaming agency, player means for receiving game information, presenting the game information to a user and providing at least one game control, a means for providing the game information over a data network to a predetermined number of receiving means, means for determining the location of the receiving means, and means for disabling the local access means. Other similar embodiments may also be comprised of means for creating an auditable record of gaming transactions on the playing means and on the gaming means.

[0015] Another embodiment of a gaming system, in addition to the features of the embodiments discussed above, may also include customized promotional messages to players of gaming devices.

[0016] On a remote player device, an embodiment of a method of remotely accessing a host gaming device may include: establishing access to the host gaming device through a data network, receiving gaming related information from the host gaming device through the data network, presenting the gaming related information to a player, receiving at least one control signal from the player, sending the control signal to the host gaming device through the data network, and disabling local use of the host gaming device. In one embodiment, the method may also include recording each gaming transaction occurring on the remote player device. Another embodiment of the method may include providing a geographic location of the remote player device. In another embodiment of the method, the age of the user of the remote player device is also provided.

[0017] On a host gaming device, an embodiment of a method of providing remote access, including: verifying the geographic location of a remote player device, establishing a gaming session on a host gaming device from a remote player device through a data network, receiving at least one control signal from the remote player device through the data network, and sending gaming related information from the gaming device through the data network. One embodiment of a method may also include recording each gaming transaction occurring on the host gaming device,

[0018] In order to provide tolerance for failures of system components, a method of resuming an interrupted gaming session on a gaming device is provided. One embodiment of a method may include generating a gaming state of the gaming session on the first gaming device, encrypting the gaming state, transporting the encrypted gaming state from the gaming device. The method may also include the converse: transporting the encrypted gaming state from the first gaming device to a second gaming device, decrypting the gaming state on the second gaming device; and loading the game state into a second gaming device to resume the gaming session.

[0019] An embodiment of a gaming system which provides for resuming interrupted gaming sessions across a data network. The system may include a first host gaming device connected to the data network, the gaming device configured to execute at least one game, generate a gaming state based on execution of at least one game, encrypt the gaming state, and send the encrypted gaming state over the data network. A second host gaming device may be connected to the data network, the second gaming device configured to receive the encrypted gaming state over the data network, decrypt the gaming state, and resume executing at least one game from the gaming state. A plurality of remote player devices, configured to receive game information provided by the host gaming device, may be connected to the data network. The gaming state may include user payment or credit information, and game jackpot or payout information.

[0020] Another embodiment of a gaming system providing resumption of interrupted gaming sessions may include means for executing at least one game, means for generating a gaming state based on execution of at least one game, means for encrypting the gaming state, and means for sending the encrypted gaming state. The system may also include means for receiving the encrypted gaming state, means for decrypting the gaming state and means for resuming executing at least one game from the gaming state.

[0021] To enable gaming regulatory compliance, methods authenticating gaming system users are also provide. An embodiment of a method of authenticating a user of a host gaming device may include receiving a security certificate from the smart card, sending the security certificate from the gaming device to an authenticator device, receiving an authentication reply from the authenticator, and playing a game in response to the authentication reply.

[0022] An embodiment of the method may also include presenting the security certificate from the gaming device to a certificate authority for authentication over a data network.

[0023] An embodiment of a method of authenticating a user of a remote player device for playing a host gaming device may include receiving an indicia of identity for a user, sending the indicia of identity to an authenticator device, receiving an authentication reply from the authenticator device, and authorizing use of a host gaming device based on the indicia of identity. The indicia of identity for a user may be provided by a biometric device, a smart card, or a password provided by the user.

[0024] Another embodiment of a gaming system provides authentication of users. The system may include a data network, a host gaming device interfaced to the data network, a plurality of remote player devices interfaced to the data network, and a security device configured to provide player credentials to at least one remote player device. The each of the remote player devices may be configured to receive game information provided by the host gaming device. The host gaming device may provide game information to a predetermined number of permitted remote

player devices. Whether a remote player device is permitted to receive gaming information may be based upon, at least in part, on player credentials provided by the security device.

[0025] In one embodiment, a method of remotely accessing a gaming device provides for creating records of gaming transactions on both host gaming devices and remote player devices sufficient to provide an auditable record for a gaming authority in the jurisdiction. The method may include establishing a gaming session on a gaming device for a remote player device through a data network, sending gaming related information from the gaming device through the data network, receiving at least one control signal from the remote player device through the data network, creating an auditable gaming session record representing each gaming transaction of a gaming session on the host gaming device and on the remote gaming device. In addition, the record may be sent to a third party, such as a gaming authority, through the data network.

[0026] In another embodiment of a gaming system, the gaming system includes a network comprised of a plurality of logical segments. A security policy controls the flow of data between logical segments. A host gaming device may be connected to the data network, the gaming device configured to execute at least one game. A plurality of remote player devices may be connected to the data network. The plurality of remote player devices are each configured to receive game information provided by the host gaming device, and to control a gaming session established on the gaming device, subject to the security policy. The security policy may be based, at least in part, on the geographic location of a logical segment.

[0027] One embodiment of the gaming system may include a promotional message server to deliver customized promotional messages to users of the gaming system. In this embodiment, a gaming system may include a data network, a promotional message server configured to provide customized promotional messages. Each message may be customized with information associated with a user of the gaming system. In addition, a gaming system may include a host gaming device interfaced to the data network, and a plurality of remote player devices interfaced to the data network. The plurality of remote player devices are each configured to receive game information provided by the host gaming device and to receive and present promotional messages.

[0028] In another embodiment, a gaming system may include a means for data communication, means for executing at least one game, means for providing game information over the data network to a predetermined number of receiving means, a plurality of means for receiving game information over the data communication means. Each means for receiving game information may be coupled to a means for receiving customized promotional messages. A gaming system may also include a means for presenting promotional messages in conjunction with gaming data.

[0029] A related method of displaying information on a remote player device is also provided. The method may include receiving a promotional message on a remote player device, presenting the promotional message in conjunction with gaming information for an amount of time; and removing the promotional message from the remote player device. Information in the promotional message may be used to calculate the amount of time to present the promotional message.

[0030] A remote player interface of a gaming system may have a number of embodiments. In one embodiment of a gaming system, the gaming system includes data network, a host gaming device interfaced to the data network, and at least one remote player device interfaced to the data network. The remote player device is configured to receive game information provided by the host gaming device. The remote player interface of the gaming system may include a video display device in communication with the remote player device and a remote control device in communication with the remote player device. The remote control device is configured to control operation of a game.

[0031] An embodiment of method of remotely accessing a gaming device may include establishing a gaming session on the host gaming device from a remote player device through a data network, receiving gaming related information from the host gaming device through the data network, presenting gaming related information to a player via a video display device, receiving at least one control signal generated by a remote control device for controlling the gaming session, and sending the control signal to the host gaming device through the data network.

Brief Description of the Drawings

[0032] FIG. 1 depicts a simplified block diagram of a gaming system according to one embodiment of the invention.

[0033] FIG. 2 depicts a simplified block diagram of system elements relating to a host gaming device of FIG. 1 according to one embodiment of the invention.

[0034] FIG. 3 depicts a simplified block diagram of system elements relating to a remote player device of FIG. 1 according to one embodiment of the invention.

[0035] FIG. 4 is a flowchart depicting the sequence of events for acknowledging command messages in a gaming system as embodied in FIG. 1.

[0036] FIG. 5 is a flowchart depicting the sequence of events for establishing a remote gaming session, playing a game, and terminating the remote gaming session in a gaming system as embodied in FIG. 1.

[0037] FIG. 6 is a flowchart depicting the sequence of events for transferring funds from a player's source of funds in the gaming system of FIG. 1.

[0038] FIG. 7 is a flowchart depicting the sequence of events for a host gaming device of FIG. 2 to connect to a network using security certificates and a certificate authority.

[0039] FIG. 8 is a flowchart depicting the sequence of events for a gaming device of FIG. 2 to build and deliver an encrypted block of data representing the complete state of the gaming device.

[0040] FIG. 9 is a flowchart depicting the sequence of events for retrieving a block of data representing the state of a gaming device from a database and loading the block into a gaming device as performed by a gaming system embodiment as in FIG. 1.

[0041] FIG. 10 is a more detailed block diagram of a gaming system as depicted in FIG. 1.

[0042] FIG. 11 is a detailed block network diagram of a portion of a gaming system as depicted in FIG. 10.

Detailed Description of the Preferred Embodiment

[0043] The following detailed description is directed to certain specific embodiments of the invention. However, the invention can be embodied in a multitude of different ways as defined and covered by the claims. In this description, reference is made to the drawings wherein like parts are designated with like numerals throughout.

[0044] In a traditional casino environment, gaming devices are generally located on a gaming floor. Gaming devices are subject to regulation by gaming regulatory agencies. Regulations may limit the locations where gaming devices may be placed and by limit users of gaming devices to those of legal age to gamble in the respective jurisdiction. Regulatory agencies for a given jurisdiction may also limit the number of licensed gaming devices provided to a licensee. Where gaming devices are physically located on a casino gaming floor, verification of whether a device is being used in its licensed location within the jurisdiction may be determined by physical inspection of the gaming floor. Further, monitoring of the gaming floor in casinos ensures that players are of legal age as set by the jurisdiction.

[0045] An embodiment of a gaming system according to the present invention allows a licensed host gaming device to be used by one or more remote player devices geographically separated from the host gaming device, but still located within the jurisdiction of a gaming authority. FIG. 1 depicts a simplified block diagram of an embodiment of a gaming system 100 according to the invention. One or more host gaming devices 160, 161, 162 are licensed gaming devices. Although three host gaming devices are shown on FIG. 1, the gaming system 100 may employ any number of host gaming devices ranging from one to thousands. For convenience of discussion, set forth below is a description of certain aspects of the host gaming device 160. It is to be appreciated that the other gaming devices may contain the following or different aspects.

[0046] A host gaming device may be any device, comprised of electronic, mechanical, or a combination of electronic and mechanical components, which is used for gaming and which affects the result of a wager by determining win or loss. A host gaming device 160 is connected to a data network 150. In the embodiment depicted in FIG. 1, the data network of gaming system 100 is comprised of three logical segments. Gaming network 150 connects each host gaming device 160 and related elements such as the database 170 and central gaming controller 180. Remote network 120 connects remote player devices 110, 111, 112 to the system. Backbone network 140 provides interconnection between the gaming network 150 and the remote network 120.

[0047] The database 170 may be computer server running database software, or any other commercially available database solution. In one embodiment, as depicted, the database 170, is a casino database. In other embodiments, the database may also contain other data related, or unrelated to the casino operation.

[0048] Remote network 120 connects remote player devices 110, 111, 112 to the system. Each remote player device 110 allows a user to play a game executing on a host gaming device 160. For convenience of discussion, set forth below is a description of certain aspects of the remote player device 110. It is to be appreciated that the other remote player devices may contain the following or different aspects. Although three remote player devices are shown on FIG. 1, the gaming system 100 may employ any number of remote player devices ranging from one to thousands.

[0049] The remote network 120 may be any form of computer network, as discussed below. In one particular embodiment, the remote network 120 is part of a network provided by a cable television system. FIG. 10 depicts an embodiment of a gaming system where the remote network 120 is provided through a digital home communications terminal (DHCT) 1000, such as a set-top box.

[0050] Each host gaming device 160 may be located in any location approved by a gaming agency, such as a casino gaming floor. A host gaming device 160 provides a legally regulated random number generator. Once generation of random number has been performed, a game result is determined. Any further interaction through the game's user interface is for the benefit of a user. For example, in one embodiment of a gaming system, the host gaming device may be a slot machine. After payment is made, through a coin, token, credit device, etc, the player pulls a lever arm to execute play. In a mechanical game, for example, a slot machine, a game result may be determined by the interaction of spinning wheels. In a host gaming device 160 of an embodiment of the present invention, however, pulling the arm triggers generation of a random number which determines the game result. Thus any spinning wheels or its electronic equivalent is

purely for entertainment of the user. A host gaming device 160 plays at least one game of chance, including, but not limited to, Slots, Blackjack, Poker, Keno, Bingo, or Lotteries.

[0051] FIG. 2 depicts a more detailed block diagram of an embodiment of a gaming system 100 showing additional gaming system elements coupled to the host gaming device 160. The host gaming device 160 may include local controls 220 such as an arm. The host gaming device 160 may have a display 210 to present the results of a game to a user. Further, the gaming device 160 may have a smart card reader 280. Functions of the smart card reader 280 may include receiving payment for a game, or identifying a user for promotional or loyalty programs. A biometric identity device 290, such as a fingerprint scanner, may be used for similar functions by the gaming system.

[0052] Networks 120, 140, 150 may include any type of electronically connected group of computers including, for instance, the following networks: Internet, Intranet, Local Area Networks (LAN) or Wide Area Networks (WAN). In addition, the connectivity to the network may be, for example, remote modem, Ethernet (IEEE 802.3), Token Ring (IEEE 802.5), Fiber Distributed Datalink Interface (FDDI) Asynchronous Transfer Mode (ATM), Wireless Ethernet (IEEE 802.11), or Bluetooth (IEEE 802.15.1). Note that computing devices may be desktop, server, portable, hand-held, set-top, or any other desired type of configuration. As used herein, the network includes network variations such as the public Internet, a private network within the Internet, a secure network within the Internet, a private network, a public network, a value-added network, an intranet, and the like. In embodiments of the present invention where the Internet is the backbone network 140, gaming network 150 and remote network 120 may form a virtual private network (VPN) transported over the Internet.

[0053] In preferred embodiments, the remote network 120 may be a closed-loop network, such as the cable network depicted in FIG. 10. A closed-loop network 120 may have a limited geographic scope which allows the geographic location of a remote player device 110 to be identified. For example, a given cable network may be limited to a specific hotel. Each hotel room may be provided with a remote player device 110 which may then be identified with that location. In other embodiments, the remote network 120 may be a mobile telephone network which is capable of identifying a caller's geographic location.

[0054] As depicted in the simplified block diagram of FIG. 3, a remote player interface 300 may comprise a remote player device 110, a display 310 for presenting game information and a control 320 to provide user game control for the remote player device 160. In one embodiment, a remote player interface 110 may also comprise a remote control 395 to provide game controls. In preferred embodiments of the remote control, the connection 394 between the remote control 395 and the remote player device 160 may be any type of wireless connection,

including infra-red based protocols, or a RF wireless protocol such as Bluetooth (802.15.1). The remote control 395 may also be connected to the remote player device 160 through a wired connection such as Universal Serial Bus (USB), serial, or equivalent connection. The remote control 395 may also include controls customized for gaming. A handheld computer may also comprise a remote control 395.

[0055] The display 310 may be a television, a personal computer, or a handheld computer device. A fixed or wireless telephone handset may comprise a display 310 and controls 320 of a remote player interface. In some embodiments the controls 320 may be integrated with display 310, as for instance, in a touch screen.

[0056] In one embodiment, the game information may be a random number which represents the result of the game, information related to gaming device jackpots, or player credits. In another embodiment, the gaming information may be multimedia, sound and images, including, in one embodiment, video, representing the execution of a game. In another embodiment, game information may also be software for execution on a remote player device 110 or on any element of a remote player interface 300, such as a remote control 395, which interactively presents the game through the remote player interface 300.

[0057] To enable regulatory conformance of the gaming system, gaming device users must be geographically within an approved jurisdiction and of legal age in the jurisdiction. In a regulated gaming environment, such as a gaming floor, physical control of the premises allows enforcement of this requirement. For remote player devices 110 not operated in the regulated gaming environment of a gaming floor, the age of the user of a remote player device 110 must be verified before game information is provided by a host gaming device 160. Credentials may be received from a user using a variety of security devices and compared to records, such as in a database 170 to confirm identity and thus age of the user.

[0058] To ensure compliance with regulatory requirements, a gaming system 100 may identify the geographic location of a remote player device 110. As discussed above, a network 120 may be a closed-loop network 120 whose devices are thereby identified in geographic location by the location of that network. Other embodiments may employ a GPS system on the remote player device 110 to provide the geographic location of the device 110. In other embodiments, the remote network 120 may be a mobile communications network which provides the geographic location of network clients, such as a remote player device 110.

[0059] In one embodiment, a security device may be a smart card reader 380 that is coupled to the remote player device 110. In embodiments using a smart card reader, a user inserts a smart card into the reader which provides credentials sufficient to verify the age of the user. In

one such embodiment, indicia present on the smart card reader are compared to records in a casino database 170 to verify the age of the user.

[0060] In other embodiments, a remote player device 110 may be coupled to a biometric identity device 390, such as a fingerprint scanner. In one embodiment, information received from the biometric identity device 390 may be compared to records in a casino database 170 to verify the age of the user. In other embodiments a biometric identity device 390 may be retinal scanner or facial recognition device.

[0061] In some embodiments, the controls 320 may include an input device (not pictured in FIG. 3) coupled to a remote player device 110 to receive a password or PIN as a security device. The password or PIN may be compared to information, such as records in a casino database 170 to verify the identity, and thus the age, of the remote player device user. For example, the input device may be a keyboard, rollerball, pen and stylus, mouse, or voice recognition system. The input device may also be a touch screen associated with an output device. The user may respond to prompts on the display by touching the screen. The user may enter textual or graphic information through the input device. The controls 320 may be coupled to a display 310 in the form of a personal computer, a television, a television with a set-top box, a handheld computer, or a telephone, fixed or mobile, handset.

[0062] Embodiments of a remote player device 110 may be a television, a cable interactive set-top box, a remote control, a personal computer, or a mobile or fixed telephone handset. Another embodiment may comprise a handheld computer coupled to a fixed or preferably wireless network. Also, a host gaming device 160 may also be a remote player device 110.

[0063] In one embodiment, a remote gaming device 110 may be in a location approved by a gaming agency with controls 320 and display 310 which match the appearance of a stand-alone gaming device. For example, a remote gaming device 110 may appear to be a slot machine with an arm control 320, a mechanical or electronic "slots" display 310. In other embodiments, remote gaming devices 110, regardless of location, may have controls and displays which match the appearance of a host gaming device 160. This may include control devices coupled to personal computers or set-top boxes which may be customized for one or more games.

[0064] Indicia of identity and age received from a smart card reader 380, biometric identity device 390, or user entry of a password may also be compared to records stored on the remote player device 110. For example, a remote player device 110 in a hotel room may be programmed by hotel staff to store identification information for eligible guests in the room containing the gaming device without the identification information being included in the casino database 170. In these embodiments, access to the remote player device thus may itself be an indicium of legal age to the central gaming controller 180 or host gaming device 160.

[0065] A central gaming controller 180 may manage the interaction of remote player devices and host gaming devices. The central gaming controller 180 may comprise one or more server computers or may be integrated with a host gaming device. In the embodiment depicted in FIG. 10, the application server 1027 and request processing servers 1023 comprise the central gaming controller 180.

[0066] One embodiment of a gaming system 100 comprises a single remote player on a remote player device 110 establishing a gaming session on a host gaming device 160 with no local player using the host gaming device 160. In this embodiment, the local controls 220 of a host gaming device 160 become disabled for local play during the remote gaming session. Correspondingly, a host gaming device 160 in this embodiment also becomes unavailable for remote play while a player uses the local controls 220 to use the host gaming device 160.

[0067] Another embodiment comprises a single player using the local controls 220 of a host gaming device 160 and a single remote player on remote player device 110 concurrently. Thus in this embodiment, the local game controls 220 on the host gaming device 160 are not disabled during the remote gaming session.

[0068] Another embodiment of the gaming system 100 comprises a single local player of the host gaming device 160 and multiple remote players on a plurality of remote player devices 110 having concurrent gaming sessions. A similar embodiment comprises multiple concurrent remote players and no local players on the host gaming device 160 because the local controls 220 may be disabled during the remote gaming sessions.

[0069] Another embodiment of a gaming system 100 comprises one or more remote player devices 110 which are physically located in a location approved by a gaming agency and networked to a host gaming device 160 that hosts both local and remote player sessions. Players physically located in the casino may occupy a remote player device 110 and play the games provided by the host gaming device 160. Concurrently, gaming sessions to one or more remote player devices 110 physically located outside the casino may be provided. Thus, in this embodiment, players may concurrently play using the host gaming device 160, a physically remote player device 110, or a remote player device 110 in a location approved by a gaming agency.

[0070] Another embodiment of the invention comprises one or more remote player devices 110, physically located in a location approved by a gaming agency and at least one host gaming device 160. In this embodiment, player sessions may only be established on a host gaming device 160 from a remote player device 110 if that remote player device 110 is physically located in a location approved by a gaming agency, such as a casino gaming floor. Players may also play the host gaming device 160 using local controls 220 concurrently with remote player sessions.

Thus, in this embodiment, players may concurrently play using the host gaming device 160, or a remote player device 110 that is located in a location approved by a gaming agency.

[0071] In each of the above disclosed embodiments, the remote player devices 110 that may concurrently receive game information from a host gaming device 160 may be limited to a predetermined number that is determined by a regulatory gaming agency for the jurisdiction.

[0072] A remote player device 110 that is physically located in the casino in a location approved by a gaming agency, such as a casino gaming floor, may differ from a remote player device physically located outside the casino floor. In one embodiment, a remote player device 110 located in a location approved by a gaming agency resembles the appearance of a stand-alone gaming device and may thus be similar in appearance and operation to the host gaming device 160.

[0073] In one embodiment, a remote player device 110 requests game data from the host gaming device 160 by sending a request for a game to a central gaming controller 180. The central gaming controller 180 then transmits the request for a game to the host gaming device 160. The host gaming device 160 receives the request and provides game data to the central gaming controller 180 that passes to the remote player device 110. That information is then translated into a game by the remote player device 110 and displayed or performed to the player. The remote player device 110 may contain on-board hardware and software that may be required to present a game. The regulated portion of hardware and software required to execute a game, such as a random number generator, is on the host gaming device 160 and the information transmitted to the remote player device 110 each time a game is requested.

[0074] Gaming devices according to an embodiment of the invention may use mixed-protocol delivery systems for game content and game results. Game information and results comprising image and sound data may be delivered by packet based network protocols such as IP datagrams, by connection-oriented network protocols, or by a combination of both. Streaming media protocols may also be employed. During a given gaming session, these communication methods may be used interchangeably or concurrently.

[0075] In one embodiment, communication over the data networks 120, 140, or 150, may use IP datagrams to package image and sound data comprising a host gaming device interface and display, encrypts it, and delivers it to the remote player device.

[0076] Internet Protocol (IP) is a network layer protocol used by many corporations, governments, and the Internet worldwide. IP is a connectionless network layer protocol that performs addressing, routing and control functions for transmitting and receiving datagrams over a network. The network layer routes packets from source to destination. An IP datagram is a data packet comprising a header part and a data part. The header part includes a fixed-length header

segment and a variable-length optional segment. The data part includes the information being transmitted over the network. As a connectionless protocol, IP does not require a predefined path associated with a logical network connection. Hence, IP does not control data path usage. If a network device or line becomes unavailable, IP provides the mechanism needed to route datagrams around the affected area.

[0077] The remote player interacts with a game through a remote player interface 300. A remote player device 110 may send commands back to the central gaming controller 180 as, in one embodiment, IP datagrams. The IP datagrams are interpreted by the central gaming controller 180 and used to proxy user interface interaction between the gaming device and the remote player. Game results may also be packaged as IP datagrams and delivered to the remote player through this method.

[0078] Alternative embodiments may use connection-oriented protocols such as TCP, or a combination of connection oriented protocols and connectionless packet protocols such as IP. Transmission Control Protocol (TCP) is a transport layer protocol used to provide a reliable, connection-oriented, transport layer link among computer systems. The network layer provides services to the transport layer. Using a two-way handshaking scheme, TCP provides the mechanism for establishing, maintaining, and terminating logical connections among computer systems. TCP transport layer uses IP as its network layer protocol. Additionally, TCP provides protocol ports to distinguish multiple programs executing on a single device by including the destination and source port number with each message. TCP performs functions such as transmission of byte streams, data flow definitions, data acknowledgments, lost or corrupt data re-transmissions, and multiplexing multiple connections through a single network connection. Finally, TCP is responsible for encapsulating information into a datagram structure.

[0079] Static content comprising the game interface or other elements of the game may be delivered to the remote player device 110 and stored on the remote player device. This delivery of content may use a mixed-protocol as described above. A static image may be a fixed image or an animation activated by the remote control device. Such images may further be overlaid with additional game content such as images and sound that is delivered dynamically during game play.

[0080] In an embodiment of the invention, a central gaming controller 180 converts image and sound data comprising the gaming device interface and display from the remote machine into a data stream (for example but not limited to MPEG-2), encrypts it, and delivers it to the remote player device 110. The remote player interacts with the game using the remote player interface 300 to send commands back to the central gaming controller as IP datagrams. The IP datagrams may be interpreted by the central gaming controller 180 and used to proxy user interface

interaction between the gaming device 160 and the remote player device 110. Game results may also be packaged as a data stream and delivered to the remote player through this method.

[0081] FIG. 4 is a flowchart depicting a method employed when a command message is acknowledged by a central gaming controller 180 according to one embodiment of a gaming system 100. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Note that in some embodiments, not all messages received by the central gaming controller 180 need be acknowledged. Starting at step 401, a command message is sent to the central gaming controller 180 by a host on the network. The host may be remote player device 110 used for remote play, or other authorized network devices. Next, at step 405, a qualified request message is received by the central gaming controller 180. Moving to step 410, the message is then recorded in a database. The database may be a casino database 170. Proceeding to step 415, the message is processed and a response prepared. Next at step 420, the response is recorded in the database. Moving to step 425, the response is sent back to the requesting device. At step 430, a test to determine whether an acknowledgment of the message has been received is made. Continuing at step 435, if the timeout value has passed control continues to step 440, if the timeout period has not expired control returns to step 430. Moving to step 440, whether the message has not been acknowledged by the originating host is tested. If acknowledgement has been received, control proceeds to 445, if not control proceeds to step 455. At step 445, the message status is recorded as "RECEIVED" and the process moves to the end state. Returning to step 455, where the process flow continues following an unacknowledged message, the system sends a status request message to the sending host. Next, at step 460, if the originating device responds to the message then flow continues to step 465, otherwise control moves to step 480. Moving to step 465, a diagnostic message is sent to query whether the originating device is ready to receive the original message. Next at step 470, if the originating host responds that it is ready to receive the original message, then control transfers to step 425 but if the originating host fails to respond then control moves to step 480. Moving to step 480, the status of the originating host is set to offline until such time as the originating host can respond or reinitializes, and the process moves to the end state.

[0082] FIG. 5 is a flowchart depicting a method used when a request for a remote gaming session is received, when playing a game, and when terminating the remote gaming session. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Starting at 510, a request for a remote gaming session is received as a request for a secured encrypted connection to the central gaming controller 180. Included in the request are the remote players security credentials in the form of a security certificate, for example, X.509 certificate. Next at 515, the security credentials are authenticated.

This authentication may be performed by submitting the security certificate to a certificate authority for authentication. Moving to 520 if the player is not authenticated, control reverts to 515. Continuing to step 525, the central gaming controller 180 establishes a secure encrypted connection with the remote player device 110. Next, at step 530, if required the player transfers funds to use during the remote gaming session. Continuing to step 535, the player then chooses a host gaming device 160 to play. Next, at step 540, in one embodiment, when a host gaming device 160 is chosen for remote access play the local controls of the host gaming device 160 is disabled to prevent local play. Moving on to step 545, a remote play session is opened on the host gaming device 160. Continuing at step 550, after a remote gaming session is established on the host gaming device, the central gaming controller 180 sends a message to the host gaming device 160 instructing it to displace representations of its user controls, graphics and sounds to the remote player interface 300. The central gaming controller 180 directs the host gaming device 160 controls over the secured encrypted connection and manages the remote gaming session. Next at step 555, the remote player may transfer funds from a player account to the host gaming device 160 for wagering on the host gaming device 160. Moving to step 560, a wager is made. Next at, 656 a game is played. Continuing to step 570, the central gaming controller 180 delivers the results of the game to the remote player interface 300. Next at step 571, the remote player may repeat the sequence from step 560. Next at step 575, if there are any credits on the host gaming device 160 when the player terminates the remote gaming session, the central gaming controller 180 automatically transfers those credits back to the players account. Moving to step 580, the central gaming controller 180 terminates the remote gaming session with the host gaming device 160. Continuing to step 585, the central gaming controller 180, enables local play on the host gaming device 160, control is then transferred to the end state.

[0083] FIG. 7 is a flowchart depicting a method for a host gaming device 160 to become connected to a network using security certificates and a certificate authority. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Starting at 705, a host gaming device 160 starts the process of connecting to a network as part of its initialization mode. Continuing to step 720, at a point during initialization, the host gaming device 160 submits a security certificate to a certificate authority for authentication. Moving to step 725, the certificate authority authenticates the certificate. Next at step 730, if the certificate is authenticated control moves to step 740, otherwise control moves to step 735. Continuing on to step 740, the host gaming device 160 is permitted onto the network and the process moves to its end state. Returning to step 735, if the certificate is not authenticated then a log entry is generated and the host gaming device 160 is not permitted onto the network.

[0084] Embodiments according to the invention may also use instant messaging and/or email messaging systems. Typical instant messaging systems permit computer users to type text messages and add file attachments into a host program and have the host program automatically deliver the text through a virtual direct connection to a target computer. Public email systems are those available for general use, as over the internet. Examples of public instant messaging systems in use today include but are not limited to chat programs like IRC, MSN Messenger, AOL Instant Messaging and a host of others. Private systems are restricted to a casino or gaming system. Typical email messaging systems permit messages and file attachments to be entered into a host program and addressed to a specific recipient on a network. These messages may not be delivered directly to the addressee, but are sent to a storage area where the recipient may retrieve the message at a time of their own choosing.

[0085] Gaming devices 160 and remote player devices 110 routinely exchange information with a central gaming controller 180 for, typically, but not limited to, account and game tracking functions. In one embodiment of the invention, devices may send and receive data over public and/or private email-type messaging systems. The message body of any particular message may vary, using a proprietary or non-proprietary format, and may be encrypted or in human-readable format. Messages may be sent at a time determined by the message originator, typically, but not exclusively, in response to an event. The recipient of the message may be any device capable of consuming the message. The message recipient may be responsible for checking the prescribed message storage area for messages addressed to it. The message recipient may reply to a received message or may generate a new message to a specific recipient, a group of recipients, or all recipients connected to the system. Remote player devices 110 may periodically check for new messages in the system and process them.

[0086] According to one embodiment of the invention, gaming devices 160 may send and receive data over public and/or private instant messaging systems. The message body of any particular message may vary, using a proprietary or non-proprietary format, and may be encrypted or in human-readable format. Messages may be sent at a time determined by the message originator, typically, but not exclusively, in response to an event. The recipient of the message may be any device capable of consuming the message. Both the gaming device 160 and the message recipient may queue incoming and outgoing messages. Queuing messages permits devices involved in instant message communications to accept new messages while processing received messages and to generate outgoing messages for delivery as system resources permit.

[0087] In another embodiment according to the invention, devices may send and receive data over public and/or private email-type messaging systems. The message body of any particular message may vary, using a proprietary or non-proprietary format, and may be encrypted or in human-readable format. Messages may be sent at a time determined by the message

originator, typically, but not exclusively, in response to an event. The recipient of the message may be any device capable of consuming the message. The message recipient may be responsible for checking the prescribed message storage area for messages addressed to it. The message recipient may reply to a received message or may generate a new message to a specific recipient, a group of recipients, or all recipients connected to the system. Gaming system devices 110 and 160 may periodically check for new messages in the system and process them.

[0088] Embodiments according to the invention may present promotional messages during remote play sessions. Messages sent may comprise instant messages for promotional information, notification of events, or other pieces of information that can be communicated electronically. Promotional messages may also include jackpot and bonus information. A promotional message server may be used to construct and send promotional messages. In one embodiment, a computer server, comprising a central gaming controller 180, may also comprise the promotional message server.

[0089] A user interface may be provided to construct message templates. These templates are then used to construct a deliverable message. Embodiments of a message template may comprise a timeout value that indicates how long the message is to be displayed, the frequency with which the message displays in relationship to other scheduled messages, a limitation value that prevents the message from being displayed too often and an expiration date after which the message is no longer used in the system. Custom graphics and display modes may also be specified for a message template, such as icons, animations, and various scrolling methods.

[0090] A remote player device 110 may present a promotional message for an amount of time determined from the contents of the promotional message. The promotional message may be presented to a user in conjunction with gaming information. The presentation may contain icons, animations, and various scrolling methods. In addition multimedia such as sound and video may be utilized.

[0091] The promotional message server may also provide a dynamic data insertion function to insert player information such as the player's name or birthday into a message prior to delivery. Dynamic data insertion may be accomplished through the use of specialized tags within the message body. When encountered, the tag characters within the message are replaced with data from a related data source. The specific tag's character sequence is associated with a specific subset of the data in the data source, such as a player's name in a data source of player information. Processing comprises reading the data source and its subsets, parsing the specialized tags from the message template, indexing the data source and replacing the tag characters with data from the data source to create a deliverable message for each item in the data source. This sequence continues until all the data in the data source has been included in messages. The messages may be delivered

as they are created or queued until all items in the data source have been used to create messages, then all messages may be sent at the same time.

[0092] In one embodiment, a gaming system 100 may comprise a card reader installed in a gaming device 280 or remote player device 380. Promotional messages may be based on information obtained about a player that is either stored on a card inserted into the card reader or by using identifying information from the card to access the casino's proprietary database systems 170.

[0093] One embodiment of the promotional message server may also provide a dynamic grouping function in which a subset of players currently gaming is selected and collected into a group. Casino operators may address a message template to this dynamic subset of current players and send a specific message or messages exclusively to that subset. These messages may be constructed using the dynamic data function. The dynamic grouping function may use criteria specified by the casino and available in the casino's proprietary database systems 170 and criteria generated by live gaming activity to establish a profile that players must meet to be selected. The criteria may comprise loyalty points the player has earned, a player's birthday, length of current gaming session, or other data that is collected by the casino on players and gaming activity.

[0094] The dynamic grouping function may be scheduled to run at time intervals determined by the casino. Each time the interval is reached the promotional gaming server searches for current players that meet the established criteria and builds a dynamic group then sends the assigned message to that group of players exclusively. The gaming devices 160, remote player device 110, card readers installed in gaming devices 280 and remote player device 380, and casino proprietary database systems 170 may provide data to search for players that meet the specified criteria and assemble them into a dynamic group.

[0095] In one embodiment of the invention, the casino may advertise a casino sponsored event. The casino may use a user interface display to construct the message and schedule its delivery start time, duration of the message e.g. number of hours, days, weeks, or months that the message will run, and specific values that weight the message's delivery interval and frequency amongst other promotional messages scheduled in the system. The style of message may also be specified, including but not limited to flashing, scrolling, scroll direction, and the use of custom graphics. The casino operator may also specify the criteria players must meet to receive the message. Once the casino operator accepts the promotional message configuration, the promotional message server may deliver the message across a network to remote player devices 110 or host gaming systems 160.

[0096] An embodiment of a gaming system 100 may provide for the electronic transfer of funds to a gaming device for the purpose of making wagers. When a player chooses a gaming device 160 to play remotely, funds are electronically transferred to the gaming device and

appear as credits on the gaming device 160. The player then uses those credits to make wagers on game outcome. When the player is finished, the system transfers any remaining credits on the gaming device back to the source of funds or to an alternate storage. Limitations on the amount of funds transferred may be set for a minimum or maximum amount transferred, a minimum or maximum amount transferred within a given time period, or a minimum or maximum amount transferred for the life of the account, or a combination of any of these. The limitation may also vary between accounts, permitting one account to have a different limitation on transfers than another. When the limitation set is reached, further transactions are prevented until the limitation is resolved. The limitation may be set voluntarily by the player, by the casino, or by a gaming authority. Limitations may be set for all players within a specific jurisdiction or for selected players only. The source of funds used by a player for remote access play may be maintained in a database located on a computer that is directly or indirectly connected to the casino network 150.

[0097] FIG. 6 is a flowchart depicting an embodiment of the invention whereby a player transfers funds from a bank account to a player account for the purpose of wagering on games. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Starting at step 601, a remote player device 110 initiates an electronic funds transfer. Continuing to step 605, the central gaming controller 180 verifies the remote players banking information. Next at step 610, if the banking information is valid, control transfers to step 620, otherwise control moves to step 615. Continuing at step 620, the remote player device 110 prompts the player to enter the amount of the transfer. Moving to step 615, the central gaming controller 180 verifies fund availability. Next at step 630, if funds are not available control moves to step 615. Otherwise, control moves to step 635, where, in a one embodiment, the central gaming controller 180 may consult a casino database 170 and determine whether the remote players total gaming activity exceed limits placed on that activity. Next at step 640, if the limit is reached control moves to step 615. Otherwise, continuing at step 645, the transfer is completed. Returning to step 615, if the players banking information is not correct, funds are not available or a transfer limit is reached, then the transaction is canceled and control transferred to the end state.

[0098] An embodiment of a gaming system 100 may record the interaction between remote players and host gaming devices 160 during remote gaming sessions for the purpose of resuming games in-progress after a communications failure. If at anytime the connection between the remote player and a gaming device becomes unavailable, the system has a sufficient record of player positions to restart the game as at the time just prior to the failure. Thus an embodiment of a gaming system may record, transfer, and reinstate on a like device an encrypted block of data representing the precise state of a particular gaming device 160 at the time that the data block is requested. The encrypted block of data is generated by the gaming device 160 and transferred

using a communication protocol. The encrypted block of data may be used to continue a game in-progress that was interrupted by a gaming device 160 failure or other system failure. In addition, the payer's wager and credit data along with gaming payout data may be included in the data block. The data may also be transported to another gaming device 160 for the purpose of completing an interrupted game or resuming a gaming session. The destination gaming device 160 receives the encrypted block of data, decrypts it, and loads the game state into its own systems, allowing a game in-progress to complete or a game session to continue.

[0099] FIG. 8 is a flowchart depicting a method for a gaming device 160 to build and deliver an encrypted block of data representing the complete state of the gaming device. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Starting at 805, a central gaming controller 180 sends a message to a host gaming device 160 to initiate the build of the encrypted data block. Continuing to step 10, the gaming device responds with an acknowledgement. Next, at step 815, the gaming device 160 begins the build process. When finished with the build and encryption process, at step 820, the gaming device saves the data block to non-volatile memory in the gaming device. Continuing to step 825, the gaming device 160 sets an indication that may be queried by the central gaming controller 180 as to the status of the build/encryption process. Moving to step 830, the central gaming controller 180 checks the gaming device's status. Next at step 835, if the build/encryption process is complete, control continues to step 840, otherwise control returns to step 830. Moving to step 840, the central gaming controller 180 retrieves the data block from the gaming device 160. Next, at step 845, when the central gaming controller 180 has retrieved the data block it saves the data block to a database. Continuing to step 850, the central gaming controller then checks the validity of the saved data block. If the data block is not verified then the central gaming controller initiates another retrieval by returning control to step 840.

[0100] FIG. 9 is a flowchart depicting a method for retrieving an encrypted block of data representing the state of a gaming device from a database and loading the encrypted block into a gaming device. Depending on the embodiment, additional steps may be added, others removed, steps merged, or the order of the steps rearranged. Starting at step 905, the central gaming controller 180 retrieves a saved encrypted data block from the database. Next at 910, the controller 180 verifies the integrity of the data block. Continuing to 915, if the data block is verified, control continues to step 925, if not control moves to step 920. Returning to the flow of control at 925, the central gaming controller 180 notifies a target gaming device 160 of an intent to upload the data block. Next, at step 930, the target gaming device 160 responds with a message indicating whether it is available for the upload. Moving to step 935, if the target device is ready control moves to step 940, if not control is diverted to step 920. Returning back to step 940, the encrypted data

block is uploaded to the target gaming device 160. Next at step 945, the target gaming device 160 verifies the encrypted data block. Moving on to step 950, if the data block was verified, the gaming device moves on to step 955, if not verified, control moves to step 920. Continuing on to step 955, the gaming device 160 initializes its state to the new state defined by the received data block and the process moves to the end state. Returning back to step 920, which is reached on error conditions, an error log entry is generated and the requesting process notified.

[0101] FIG. 10 is a block diagram depicting one embodiment of a gaming system according to the present invention wherein the host gaming devices 160 are available for remote play over a network that connects to a cable modem termination system. The cable modem termination system 1005 is located at the head-end of a cable television provider who makes broadband network connectivity available as a service to its customers. Cable television customers who subscribe to broadband or digital television services access the remote network 120 through a digital home communications terminal (DHCT) 1000. The remote player device 110 may be a stand-alone cable modem or a set-top box that includes a cable modem and a digital television broadcast decoder. The DHCT 1000 may, in some embodiments include the remote player device 110. The remote player interface 300 may be any device or combination of devices that remote players operate to interact with the remote player device 110, for example, a television with remote control or a personal computer. To connect to the central gaming controller 180, a remote player uses the remote player device 110 to send messages, using, in one embodiment, IP datagrams, through the DHCT and the cable modem termination system 1005. The cable modem termination system 1005 uses a network router 1004 to route the IP datagrams over a network connection 140 to the central gaming controller 180. The backbone network connection 140 can be any type of network connection such as a dedicated T1 or fiber optic over which network traffic can be exchanged. In preferred embodiments the backbone network 140 is part of a closed loop network. However, in other embodiments, a public network such as the Internet may form at least a portion of the backbone network. Encryption of the data may be performed, either at the endpoints such as remote player device 110, at a host gaming device 160, at a central gaming controller 180, over network 120, or only over network 140.

[0102] Network traffic from the remote network 120 and backbone network 140 travels over a number of virtual local area networks (VLAN) configured using a multilayer network switch 1022. Segmenting the internal network into VLANs creates security zones whereby only permitted network traffic appears on a given VLAN.

[0103] IP datagrams are received over the backbone network 140 through network router 1020 and firewall 1021. Network router 1020 filters IP datagrams that are not coded with the configured port for access to the gaming network 150. If an IP datagram passes the network

router 1020 it then must pass the firewall 1021 in order for the IP datagram to be processed by the request processing server(s) 1023 which comprise a portion of a central gaming controller 180 in this embodiment.

[0104] The firewall 1021 has two network interfaces 1050, 1051; the external-facing network interface 1050 is connected to the router 1020 and the internal-facing network interface 1051 is connected to the multilayer network switch 1022. In this configuration the firewall 1021 acts as a type of network switch that may perform additional security checks on the IP datagram, then move the datagram to the internal-facing network interface 1051 where the multilayer network switch 1022 moves the datagram to the VLAN where request processing server(s) 1023 are located.

[0105] Each request processing server 1023 has two network interfaces 1052, 1053, both connected to the multilayer network switch 1022. Each network interface 1052, 1053 may be configured on a different VLAN of the multilayer network switch 1022. The multilayer network switch 1022 moves IP datagrams between the firewalls 1021 internal-facing network interface 1051 and the request processing server(s) 1023 external-facing network interface 1052. This embodiment provides a layer of protection for the host gaming devices 160 in the event that the request processing server(s) 1023 are compromised.

[0106] When an IP datagram arrives at a request processing servers 1023 external-facing network interface 1052, the request processing server 1023 interprets the IP datagram and issues commands over its internal-facing network interface 1053 to the application server 1027. The request processing server 1023 may reject invalid commands or make other determinations as to the appropriateness of a request that prevent the request from being passed on to the application server 1027. Likewise, the request processing server 1023 may request data from the application server for use in building its own response to the request, which may or may not require an acknowledgement from the remote player device 110 as described below.

[0107] Command messages received by the application server 1027 may be recorded in a database using the database server 1025. The application server 1027 then executes the command, which may include any function relevant to the operation of the host gaming device 160 and may or may not return data to the request processing server 1023 for delivery to the remote access player. In one embodiment, the database server 1025 may comprise the casino database 170. In other embodiments the database server 1025 and the application server 1027 may comprise the casino database 170.

[0108] Some commands may require the remote player device 110 to acknowledge the receipt of information sent from the central gaming controller 180. For commands that require acknowledgement, the central gaming controller 180 queues the status of the messages that are sent to the remote player device 110. The status of messages sent but not acknowledged is stored in a

database as "open" using the database server 1025. When the remote player device 110 receives the message it sends an acknowledgment message back to the central gaming controller, which in turn marks the message in the database as "closed"; indicating that the message has reached its destination and has been acknowledged. If the message is not acknowledged within a specified timeout, the message is resent. FIG. 4 depicts the sequence of events for the receipt, queuing and response loop for qualifying messages.

[0109] Recording of messages between the remote player device 110 and a host gaming device 160 by the central gaming controller 180 allows each game or transaction, on both the host gaming device 160 and remote player device 110, to be recorded. This allows each host gaming device or remote player device to be individually auditable using standard accounting practices in the gaming jurisdiction where the game is located. In one embodiment, a third party, such as a gaming authority may be sent the records of games and transactions online by the gaming system 100.

[0110] When the application server 1027 receives a command request that requires communication with gaming devices 160, 161, 162 it connects to those devices using terminal server 1035. Terminal server 1035 provides Ethernet connectivity to the RS232 serial interface 1054 of the game. Through that interface the remote player device 110 communicates to the gaming devices 160, 161, 162 using a communications protocol supplied by the gaming machine manufacturer. The protocol includes commands that permit the remote operation of the gaming devices 160, 161, 162 and the reporting of game results so that the application server 1027 can control remote play.

[0111] FIG. 11 depicts a more detailed network diagram of one embodiment of network 150 and elements of a gaming system 100 connected to network 150. This includes a host gaming device 160, and a database 160. As in the embodiment of FIG. 10, a central gaming controller 180 may be comprised of request processing servers 1027 and an application server 1023 connected to one or more VLANs of network 150.

[0112] While the above detailed description has shown, described, and pointed out novel features of the invention as applied to various embodiments, it will be understood that various omissions, substitutions, and changes in the form and details of the device or process illustrated may be made by those skilled in the art without departing from the spirit of the invention. The scope of the invention is indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

WHAT IS CLAIMED IS:

1. A gaming system comprising:
 - a data network, wherein the data network is comprised of at least one logical segment, wherein at least one logical segment is a closed-loop network;
 - a host gaming device connected to the data network, the gaming device configured to execute at least one game wherein the host gaming device in a location approved by a gaming agency;
 - a plurality of remote player devices connected to the closed-loop network; and
 - a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device and on each of the plurality of remote player devices,wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device, and
wherein the host gaming device is configured to provide game information to a predetermined number of permitted remote player devices.
2. A gaming system comprising:
 - a data network;
 - a host gaming device connected to the data network, the gaming device configured to execute at least one game; and
 - a plurality of remote player devices connected to the data network,wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device,
wherein the host gaming device is configured to provide game information to a predetermined number of permitted remote player devices, and
wherein at least one of the plurality of remote player devices is permitted based upon, at least in part, the geographic location of the remote player device.
3. The system of Claim 2, wherein the predetermined number is determined by a gaming agency.
4. The system of Claim 2, wherein at least one of the plurality of remote player devices is permitted based upon, at least in part, an age of a user of the remote player device.
5. The system of Claim 2, wherein the data network is, at least in part, the Internet.
6. The system of Claim 2, wherein the data network is comprised of at least one logical segment.
7. The system of Claim 6, wherein at least one logical segment is a closed-loop network.

8. The system of Claim 6, wherein the host gaming device is configured to identify the geographic location of a remote player device based, at least in part, on a logical segment corresponding to the remote player device.
9. The system of Claim 2, wherein the host gaming device is configured to identify the geographic location of a remote player device based, at least in part, on information provided by a mobile communications network.
10. The system of Claim 2, wherein the host gaming device is configured to identify the geographic location of a remote player device based, at least in part, on information provided by a GPS device.
11. The system of Claim 2, wherein the data network is, at least in part, the casino intranet.
12. The system of Claim 2, wherein the data network is, at least in part, the hotel intranet.
13. The system of Claim 2, wherein the data network is, at least in part, a wireless network.
14. The system of Claim 2, wherein the host gaming device is in a location approved by a gaming agency.
15. The system of Claim 2, wherein the host gaming device includes at least one game control configured to provide local use.
16. The system of Claim 15, wherein the host gaming device is configured to disable local use when the host gaming device is providing game information to a remote player device.
17. The system of Claim 2, wherein each of the remote player devices is in a location approved by a gaming agency.
18. The system of Claim 2, further comprising:
 - a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.
19. The system of Claim 2, further comprising:
 - a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.
20. The system of Claim 2, wherein the gaming information is, at least in part, software.
21. The system of Claim 2, wherein at least one remote player device is coupled to a credential device configured to receive information relating to a user of the remote player device.
22. The system of Claim 21, wherein the information relating to the user is an age of the user.

23. The system of Claim 21, wherein the information relating to a user is a password that is input by the user.
24. The system of Claim 21, wherein the credential device is an input device configured to receive a password from the user.
25. The system of Claim 21, wherein the credential device is a smart card reader.
26. The system of Claim 21, wherein the credential device is a biometric device.
27. The system of Claim 28, wherein the biometric device is a fingerprint reader.
28. The system of Claim 21, further comprising: a database configured to provide information associated with each of a plurality of users of the gaming system.
29. The system of Claim 28, wherein the information associated with a user includes a password.
30. The system of Claim 28, wherein the information associated with a user includes an age of the user.
31. The system of Claim 28, wherein the information associated with a user includes information relating to a fingerprint of the user.
32. The system of Claim 2, wherein the host gaming device is configured to encrypt the game information.
33. The system of Claim 2, wherein the game information is provided via a public email system.
34. The system of Claim 2, wherein the game information is provided via a private email system.
35. The system of Claim 2, wherein the game information is provided through a public messaging system.
36. The system of Claim 2, wherein the game information is provided through a private messaging system.
37. A gaming system comprising:
 - a data network;
 - a host gaming device in a location approved by a gaming agency connected to the data network, the gaming device configured to execute at least one game; and
 - a plurality of remote player devices connected to the data network.wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device, and
wherein the host gaming device is configured to disable local use of the gaming device when providing game information to the remote player devices.

38. The system of Claim 37, further comprising:
a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.
39. The system of Claim 37, further comprising:
a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.
40. The system of Claim 37, wherein the host gaming device is configured to allow no more than a predetermined number of remote player devices to concurrently receive game information provided by the host gaming device.
41. A gaming system comprising:
gaming means for executing at least one game, the game providing game information during execution;
local access means for providing local access to the game information for a user in a location approved by a gaming agency;
player means for receiving game information, presenting game information and providing at least one game control;
means for providing the game information over a data network to a predetermined number of receiving means;
means for determining the location of the receiving means; and
means for disabling the local access means.
42. The system of Claim 41, further comprising:
a means for creating an auditable record of gaming transactions on the gaming means.
43. The system of Claim 41, further comprising:
a means for creating an auditable record of gaming transactions on the playing means.
44. The system of Claim 41, wherein the predetermined number is determined by a gaming agency.
45. The system of Claim 41, further comprising:
means for receiving information associated with a user of the gaming system.
46. The system of Claim 45, wherein the information associated with the user includes the age of the user.
47. The system of Claim 45, wherein the means for receiving information associated with a user is a smart card reader.
48. The system of Claim 45, wherein the means for receiving information associated with a user is a biometric identity device.

49. The system of Claim 45, wherein the means for receiving information associated with a user is a keyboard configured to receive a password.
50. The system of Claim 45, wherein the user information includes, at least, a credential for authentication of the user.
51. The system of Claim 50, further comprising:
means for authenticating the credential coupled to means for limiting access to the gaming system.
52. A method of remotely accessing a host gaming device on a remote player device comprising:
establishing access to the host gaming device from the remote player device through a data network;
receiving gaming related information from the host gaming device through the data network;
presenting the gaming related information to a player;
receiving at least one control signal from the player;
sending the control signal to the host gaming device through the data network; and
disabling local use of the host gaming device.
53. The method of Claim 52, further comprising:
recording each gaming transaction occurring on the remote player device.
54. The method of Claim 52, further comprising:
providing a geographic location of the remote player device.
55. The method of Claim 52, further comprising:
providing information relating to a user of the remote player device to the gaming device.
56. The method of Claim 55, wherein the information relating to a user includes, at least, the age of the user.
57. The method of Claim 52, further comprising:
allowing no more than a predetermined number of remote player devices to concurrently establish a gaming session on the gaming device.
58. A method of providing remote access to a host gaming device comprising:
verifying a geographic location of a remote player device;
establishing a gaming session on a host gaming device from a remote player device through a data network;
receiving at least one control signal from the remote player device through the data network;

- sending gaming related information from the gaming device through the data network;
59. The method of Claim 58, further comprising:
recording each gaming transaction occurring on the host gaming device.
60. The method of Claim 58, further comprising:
receiving information relating to a user of the remote player device on the gaming device.
61. The method of Claim 60, wherein the information relating to a user includes, at least, the age of the user.
62. The method of Claim 58, further comprising:
disabling local access to the gaming device.
63. The method of Claim 58, further comprising:
allowing no more than a predetermined number of remote player devices to concurrently establish a gaming session on the gaming device.
64. A method of resuming an interrupted gaming session on a first host gaming device comprising:
generating a gaming state of the gaming session on the first gaming device;
encrypting the gaming state;
transporting the encrypted gaming state from the first gaming device;
transporting the encrypted gaming state to a second gaming device;
decrypting the gaming state on the second gaming device; and
loading the game state into a second gaming device to resume the gaming session.
65. A gaming system comprising:
a data network;
a first host gaming device connected to the data network, the gaming device configured to:
execute at least one game,
generate a gaming state based on execution of at least one game;
encrypt the gaming state; and
send the encrypted gaming state over the data network;
a second host gaming device connected to the data network, the gaming device configured to:
receive the encrypted gaming state over the data network;
decrypt the gaming state;
resume executing at least one game from the gaming state; and
a plurality of remote player devices connected to the data network,

wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device.

66. The system of Claim 65, wherein the remote player devices are each configured to receive an encrypted gaming state from a first gaming device over the data network and to send the encrypted gaming state to the second gaming device.

67. The system of Claim 66, wherein the first gaming device is the second gaming device.

68. The system of Claim 65, wherein the second gaming device is configured to receive an encrypted gaming state from a first gaming device over the data network.

69. The system of Claim 65, wherein the gaming state includes user payment information.

70. The system of Claim 65, wherein the gaming state includes gaming machine payout information.

71. The system of Claim 65, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.

72. The system of Claim 65, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.

73. A gaming system comprising:

means for executing at least one game;

means for generating a gaming state based on execution of at least one game;

means for encrypting the gaming state;

means for sending the encrypted gaming state;

means for receiving the encrypted gaming state;

means for decrypting the gaming state; and

means for resuming executing at least one game from the gaming state.

74. The system of Claim 73, wherein the gaming state includes user payment information.

75. The system of Claim 73, wherein the gaming state includes gaming machine payout information.

76. The system of Claim 73, further comprising:

a means for creating an auditable record of gaming transactions on the host gaming device.

77. The system of Claim 73, further comprising:
a means for creating an auditable record of gaming transactions on each of the plurality of remote player devices.
78. A method of authenticating a user of a host gaming device comprising:
receiving a security certificate from the smart card;
sending the security certificate to a certificate authority for authentication;
receiving an authentication reply from the authority; and
playing a game in response to the authentication reply.
79. A method of authenticating a user of a remote player device comprising:
receiving an indicia of identity for a user;
sending the indicia of identity to an authenticator device;
receiving an authentication reply from the authenticator device; and
authorizing use of a host gaming device based on the indicia of identity
80. The method of Claim 79, wherein the indicia of identity for a user is provided by a biometric identity device.
81. The method of Claim 79, wherein the indicia of identity for a user is provided by a password input by the user.
82. The method of Claim 79, wherein the indicia of identity for a user is provided by a smart card.
83. A gaming system comprising:
a data network;
a host gaming device interfaced to the data network;
a plurality of remote player devices interfaced to the data network; and
a security device configured to provide player credentials to at least one remote player device,
wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device,
wherein the host gaming device is configured to provide game information to a predetermined number of permitted remote player devices, and
wherein at least one of the plurality of remote player devices is permitted based upon, at least in part, on player credentials provided by the security device.
84. The system of Claim 83, wherein the security device is a smart card reader.
85. The system of Claim 83, wherein the security device is a biometric device.
86. The system of Claim 83, wherein the security device is an input device.
87. The system of Claim 86, wherein the player credentials are, at least in part, a password.

88. The system of Claim 83, wherein the remote player device is authorized to receive game information provided by the host gaming device based, in part, on the player credentials.

89. The system of Claim 83, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.

90. The system of Claim 83, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.

91. A method of remotely accessing a gaming device comprising:

establishing a gaming session on a gaming device for a remote player device through a data network;

sending gaming related information from the gaming device through the data network;

receiving at least one control signal from the remote player device through the data network.

creating an auditable gaming session record representing each gaming transaction of a gaming session on the host gaming device;

creating an auditable gaming session record representing each gaming transaction of a gaming session on the remote gaming device; and

sending the record to a third party through the data network.

92. The method of Claim 91 wherein the third party is a gaming authority.

93. A gaming system comprising:

a data network comprised of a plurality of logical segments wherein a security policy controls the flow of data between logical segments;

a host gaming device connected to the data network, the gaming device configured to execute at least one game; and

a plurality of remote player devices connected to the data network,

wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device, and

wherein the plurality of remote player devices are each configured to control a gaming session established on the gaming device subject to the security policy wherein the security policy is based, at least in part, on the geographic location of a logical segment.

94. The system of Claim 93, further comprising:

a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.

95. The system of Claim 93, further comprising:
a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.
96. A gaming system comprising:
a data network;
a promotional message server configured to provide customized promotional messages wherein each message is customized with information associated with a user of the gaming system;
a host gaming device interfaced to the data network; and
a plurality of remote player devices interfaced to the data network,
wherein the plurality of remote player devices are each configured to receive game information provided by the host gaming device and to receive and present promotional messages.
97. The system of Claim 96, wherein the remote player devices are in a location approved by a gaming agency.
98. The system of Claim 96, further comprising:
a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.
99. The system of Claim 96, further comprising:
a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.
100. The system of Claim 96, wherein promotional message are comprised of bonus information.
101. The system of Claim 96, wherein promotional message are comprised of jackpot information.
102. The system of Claim 96, further comprising: at least one database configured to provide information associated with a plurality of users of the gaming system.
103. The system of Claim 96, wherein each of the plurality of remote game devices is associated with a user.
104. The system of Claim 96, further comprising a smart card reader configured to provide information associated with a user of the gaming system.
105. The system of Claim 102, wherein the database is configured to provide information which forms, at least in part, the content of the promotional message.
106. The system of Claim 96, wherein each of the plurality of remote player devices is configured to receive and present the promotional message in conjunction with game information provided by the host gaming device.

107. The system of Claim 106, wherein each of the plurality of remote player devices is configured to present the promotional message for an amount of time.
108. The system of Claim 106, wherein the amount of time is based, at least, in part on information associated with the promotional message.
109. The system of Claim 102, wherein the database is configured to provide information which comprises, at least in part, the content of the promotional message.
110. The system of Claim 96, wherein the promotional messages are transported via an instant messaging system.
111. The system of Claim 96, wherein the promotional messages are transported via an email system.
112. A method of displaying information on a remote player device comprising:
receiving a promotional message on a remote player device;
presenting the promotional message in conjunction with gaming information for an amount of time; and
removing the promotional message from the remote player device.
113. The method of Claim 112, further comprising
calculating the amount of time based, at least in part, on information associated with the promotional message.
114. A gaming system comprising:
means for data communication;
means for executing at least one game;
means for providing game information over the data network to a predetermined number of receiving means; and
a plurality of means for receiving game information over the data communication means, each coupled to a means for receiving customized promotional messages.
115. The method of Claim 114, further comprising:
means for presenting customized promotional messages in conjunction with game information.
116. The method of Claim 114, further comprising:
means for sending promotional messages.
117. The method of Claim 114, further comprising:
means for providing data used to select which players receive customized promotional messages.
118. The method of Claim 114, further comprising:
means for providing data which forms, at least in part, the content of promotional messages.

119. The system of Claim 114, further comprising:
a means for creating an auditable record of gaming transactions on the host gaming device.
120. The system of Claim 114, further comprising:
a means for creating an auditable record of gaming transactions on each of the plurality of remote player devices.
121. A gaming system comprising:
a data network;
a host gaming device interfaced to the data network;
at least one remote player device interfaced to the data network;
a video display device in communication with the remote player device; and
a remote control device in communication with the remote player device,
wherein the remote player device is configured to receive game information provided by the host gaming device and the remote control device is configured to control operation of a game.
122. The system of Claim 121, wherein the video display device is a television.
123. The system of Claim 121, wherein the video display device is a computer.
124. The system of Claim 121, wherein the video display device is a control device.
125. The system of Claim 121, wherein the remote player device is coupled to a cable television system.
126. The system of Claim 121, wherein the data network is, at least in part, the Internet.
127. The system of Claim 121, wherein the data network is, at least in part, the casino intranet.
128. The system of Claim 121, wherein the data network is, at least in part, the hotel intranet.
129. The system of Claim 121, wherein the data network is, at least in part, a wireless network.
130. The system of Claim 121, further comprising:
a central gaming controller configured to create an auditable record of gaming transactions on the host gaming device.
131. The system of Claim 121, further comprising:
a central gaming controller configured to create an auditable record of gaming transactions on each of the plurality of remote player devices.

132. A method of remotely accessing a host gaming device comprising:
- establishing a gaming session on the host gaming device from a remote player device through a data network;
 - receiving gaming related information from the host gaming device through the data network;
 - presenting gaming related information to a player via a video display device;
 - receiving at least one control signal generated by a remote control device for controlling the gaming session; and
 - sending the control signal to the host gaming device through the data network.
133. The method of Claim 132, further comprising:
- recording each gaming transaction occurring on the remote player device.

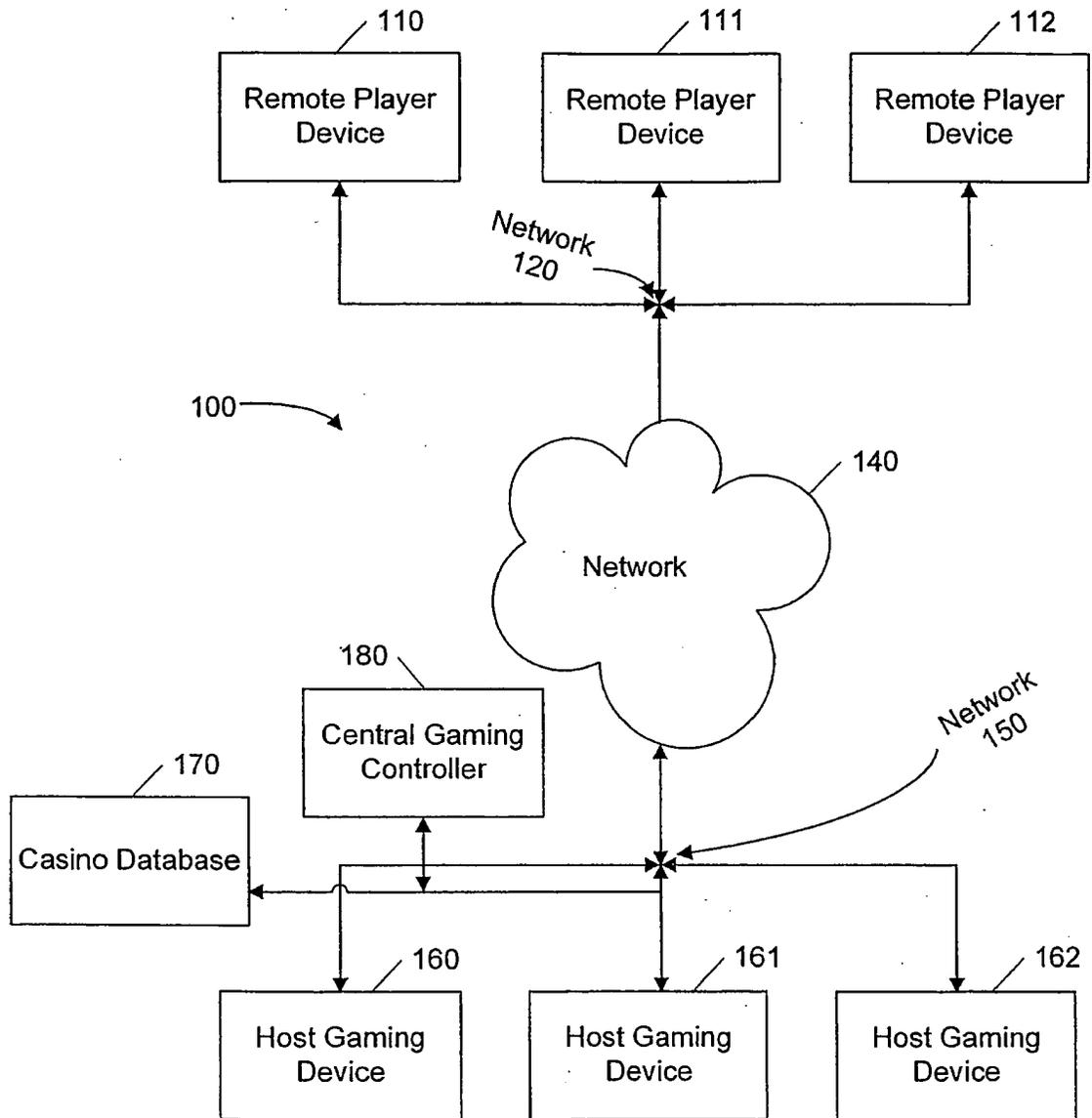


FIG. 1

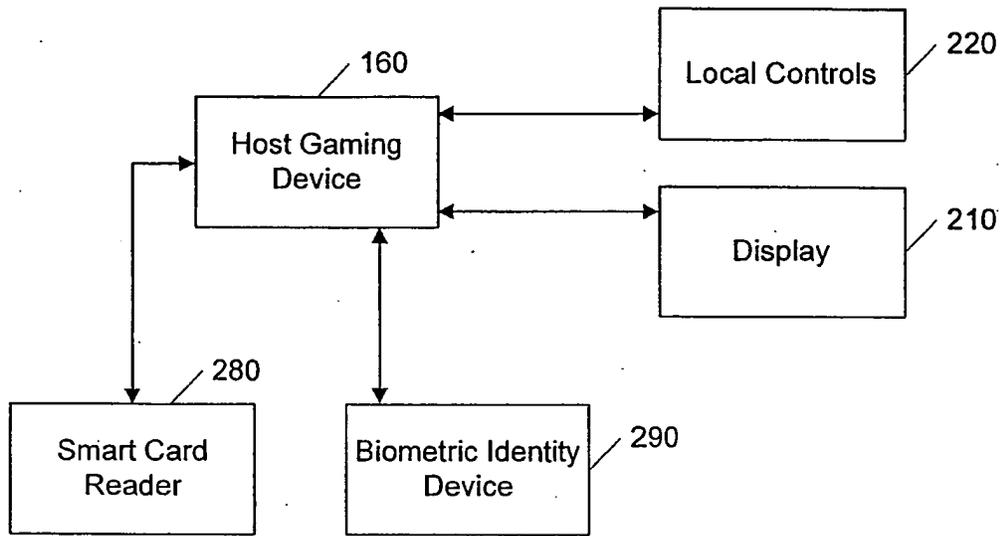


FIG. 2

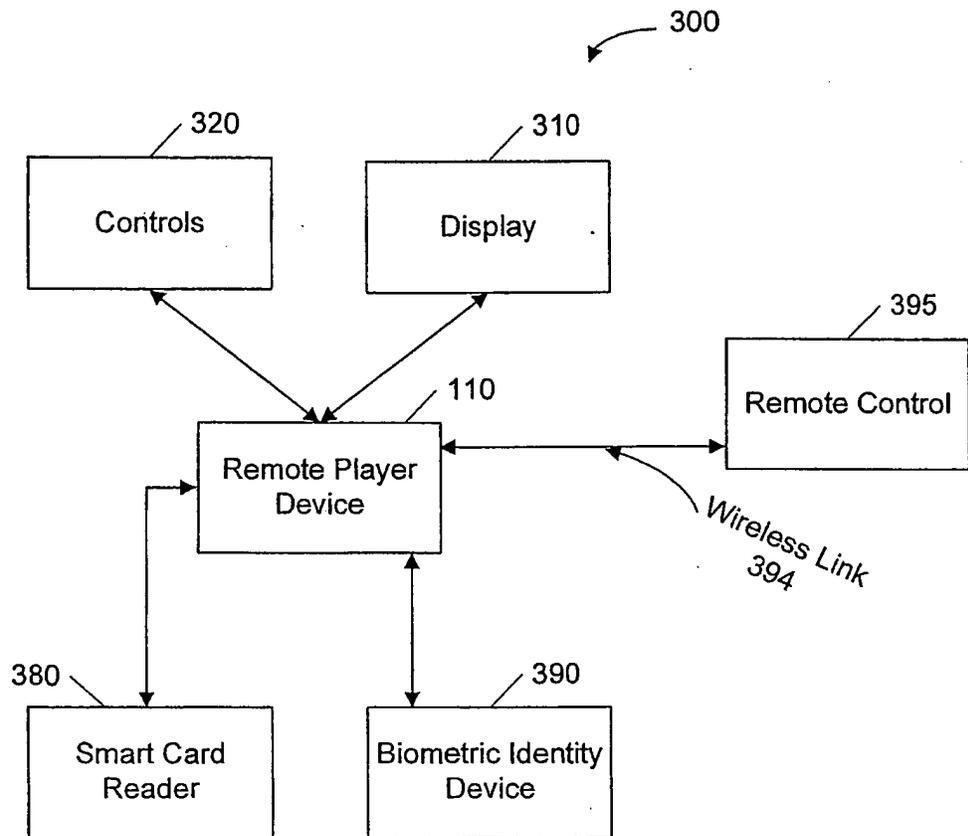


FIG. 3

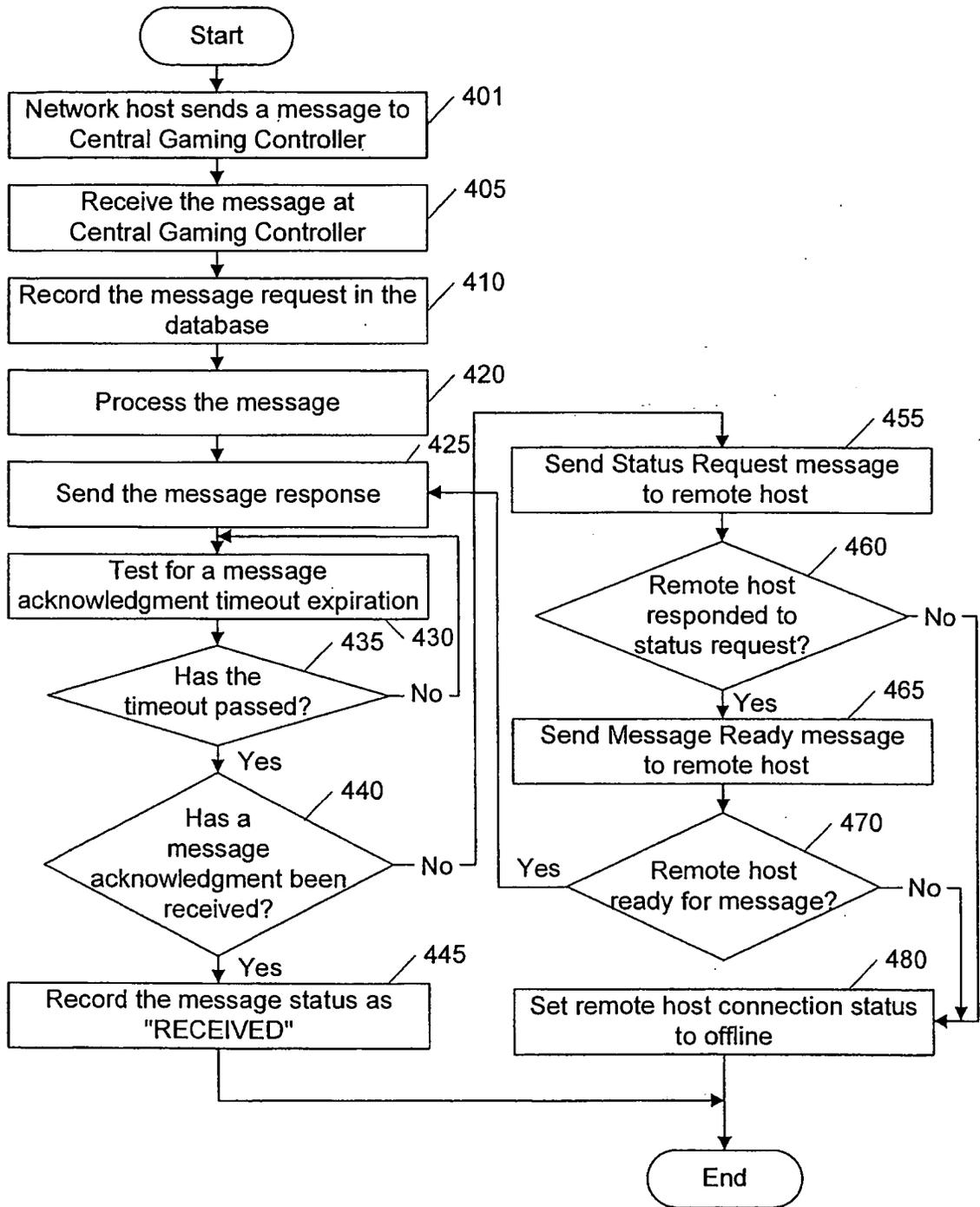


FIG. 4

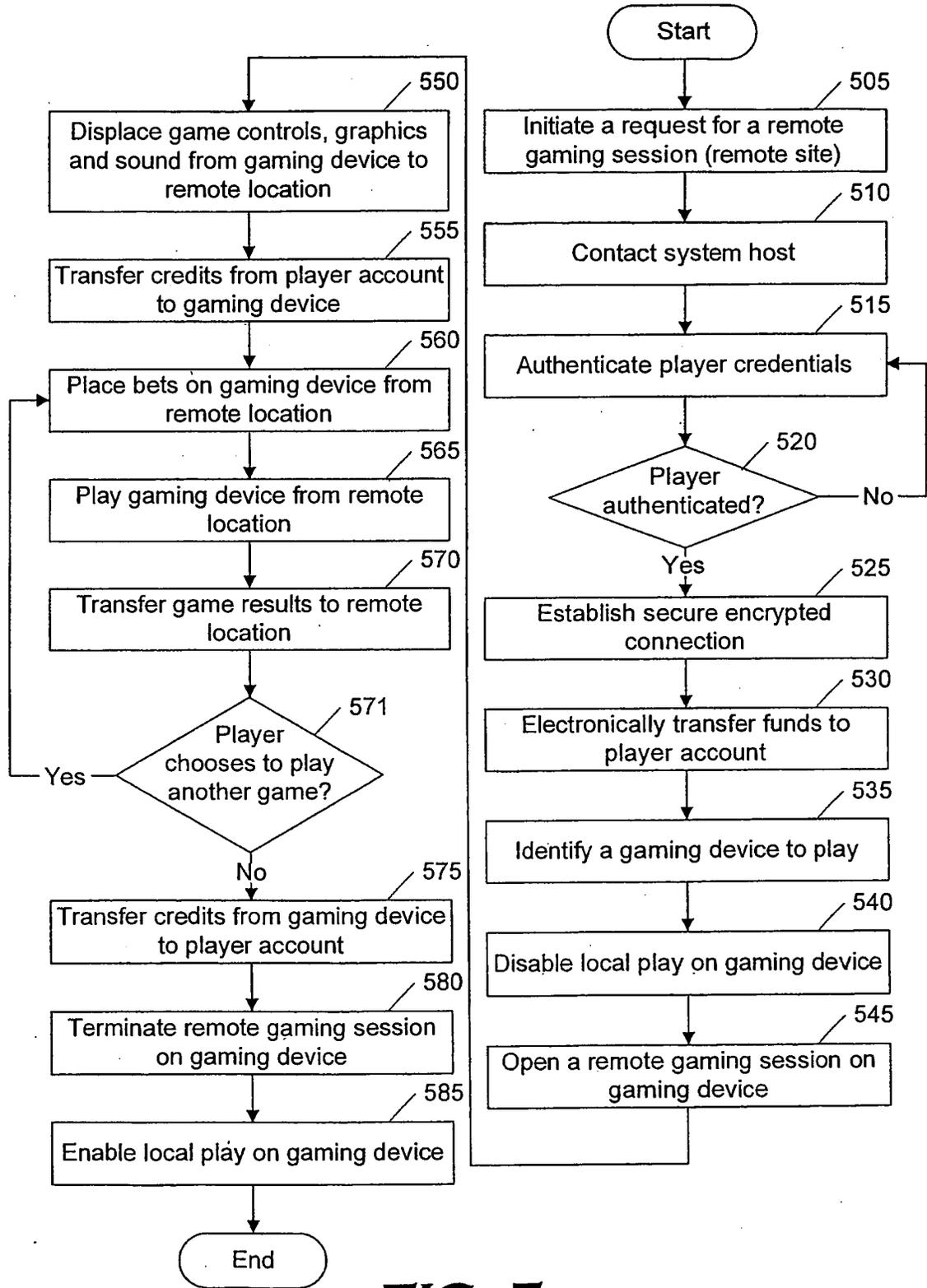


FIG. 5

6 / 11

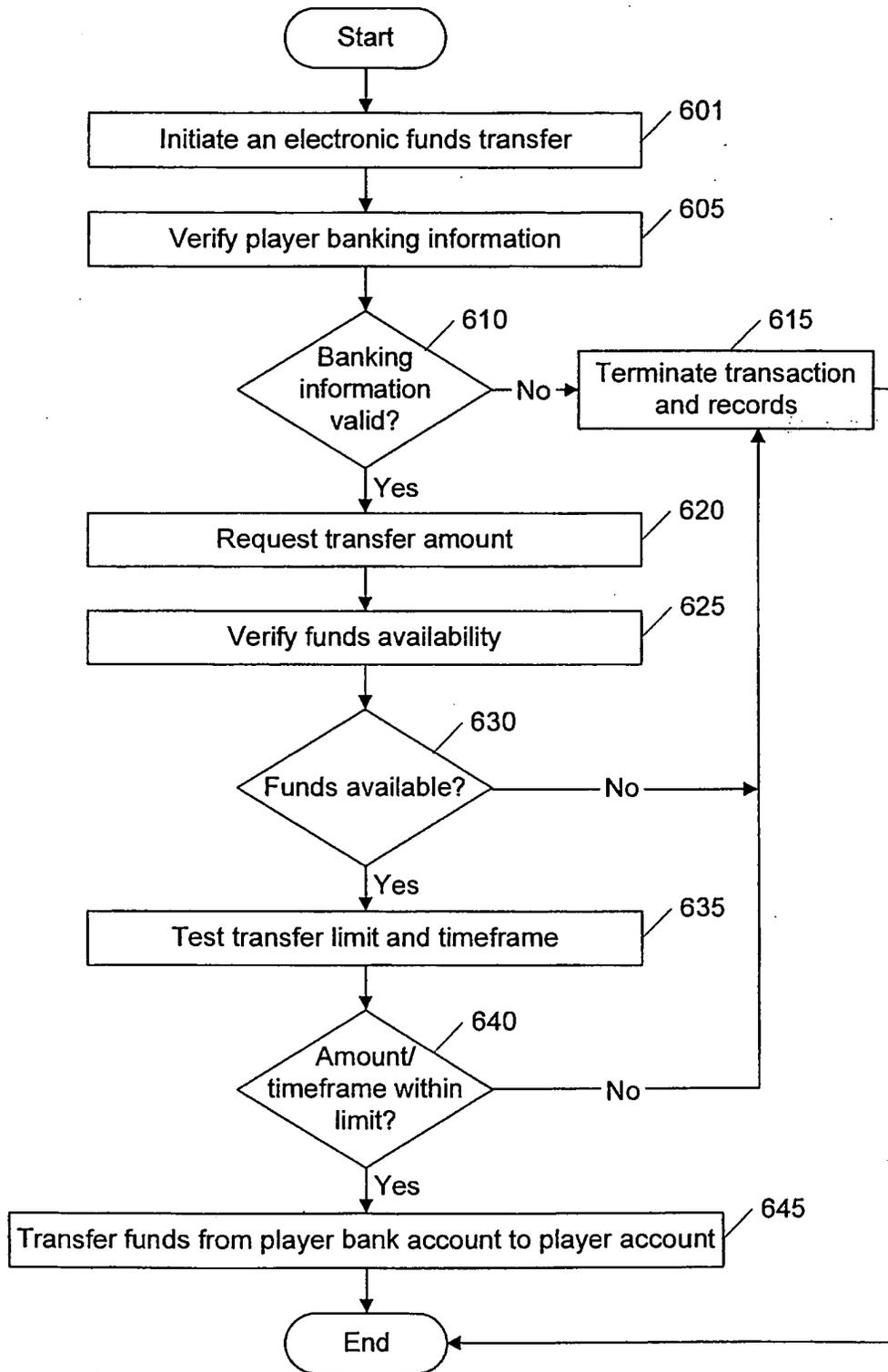


FIG. 6

7 / 11

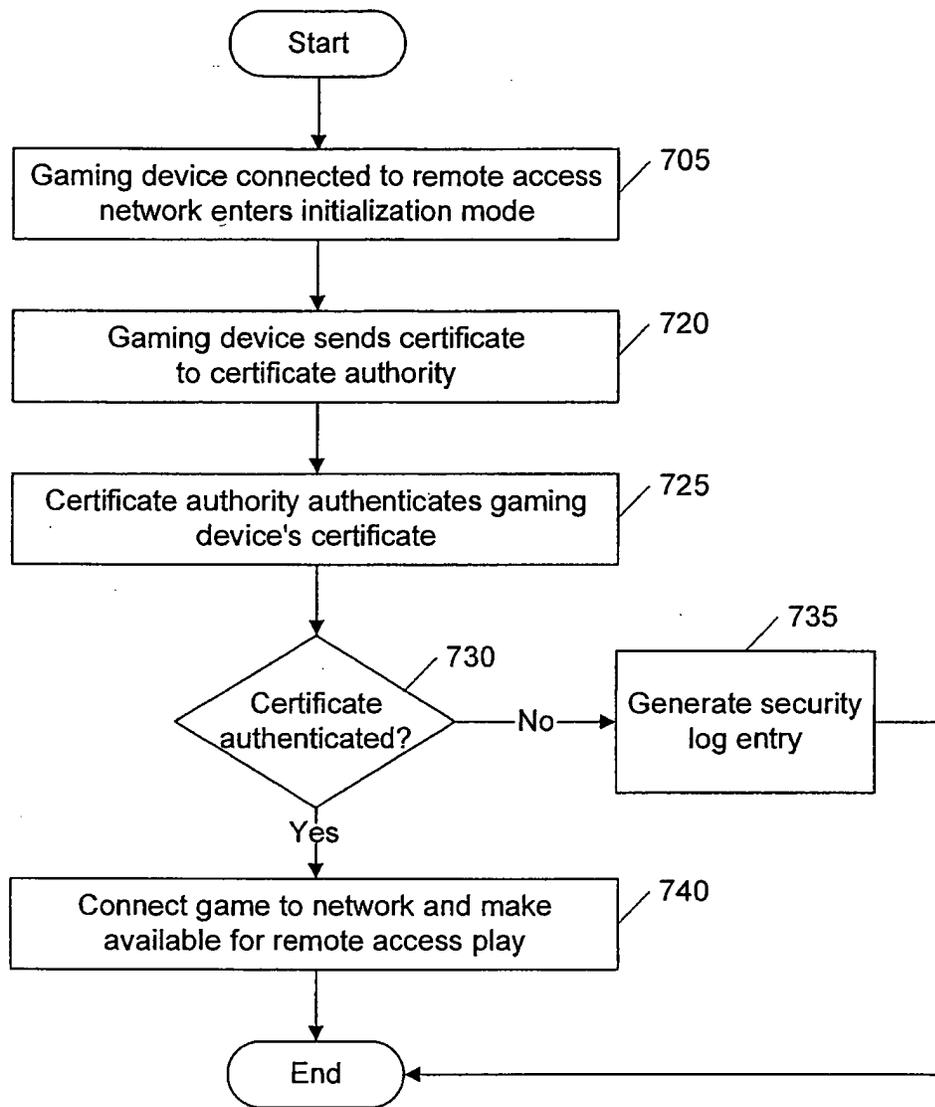


FIG. 7

8 / 11

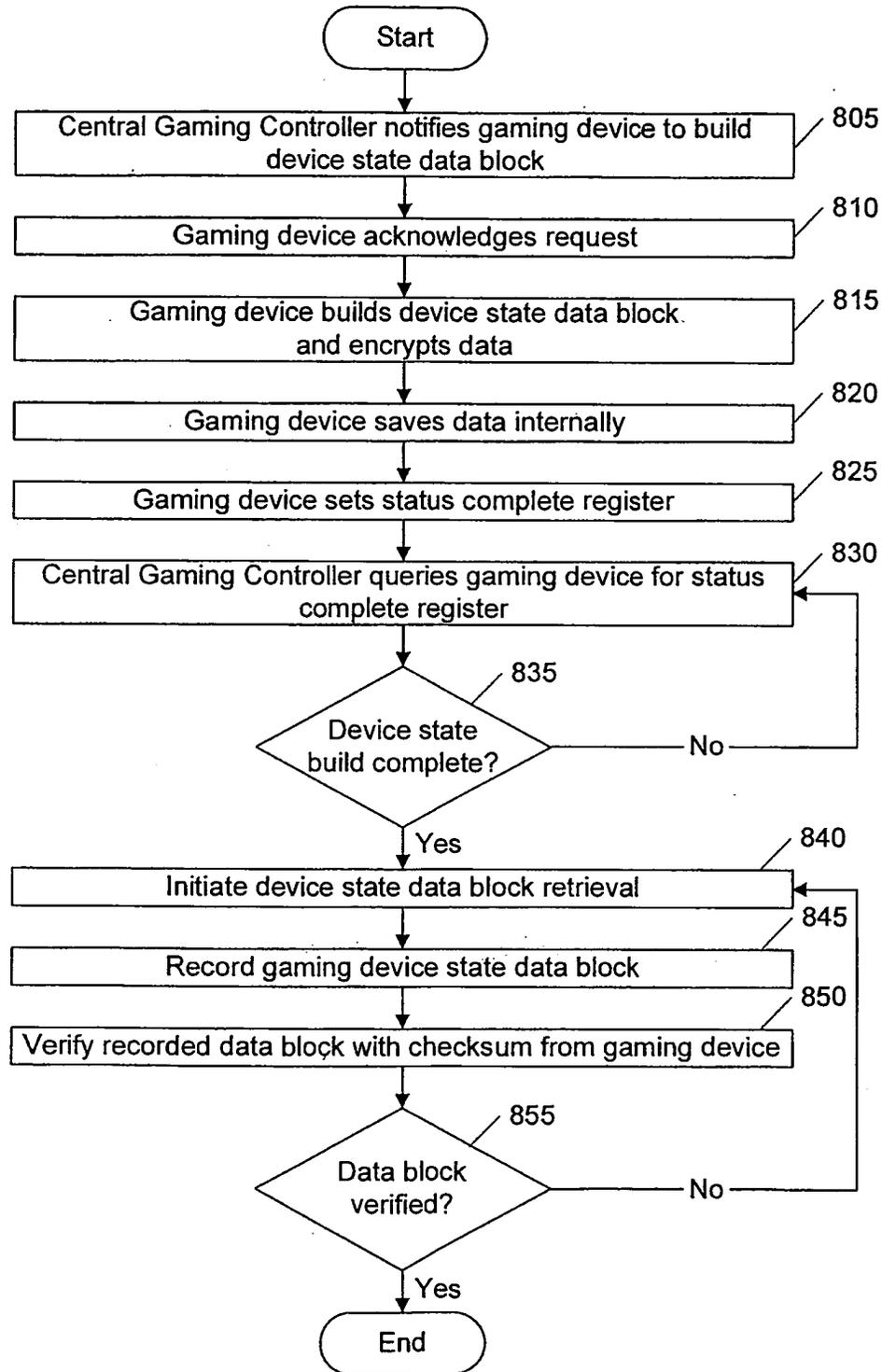


FIG. 8

9 / 11

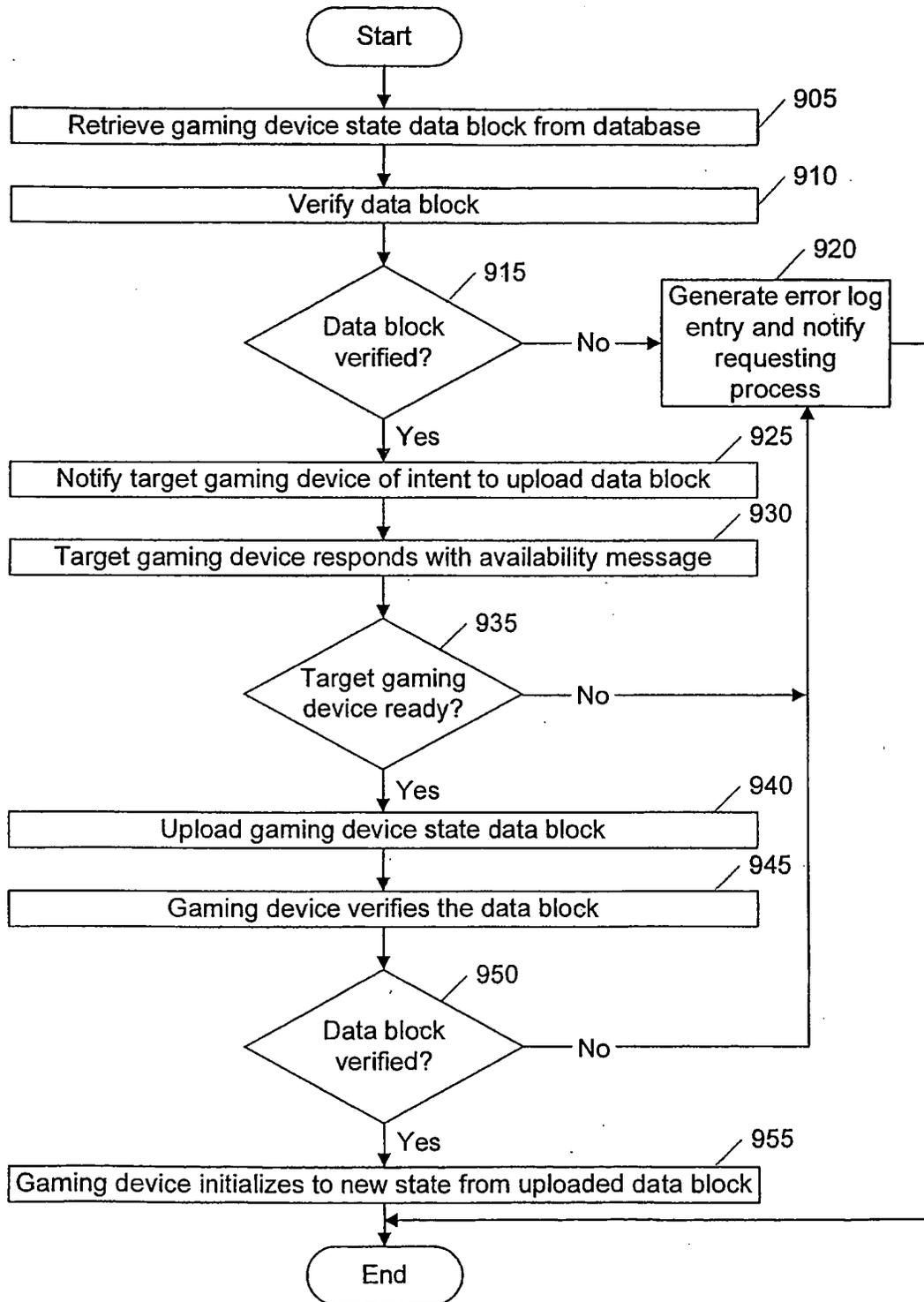


FIG. 9

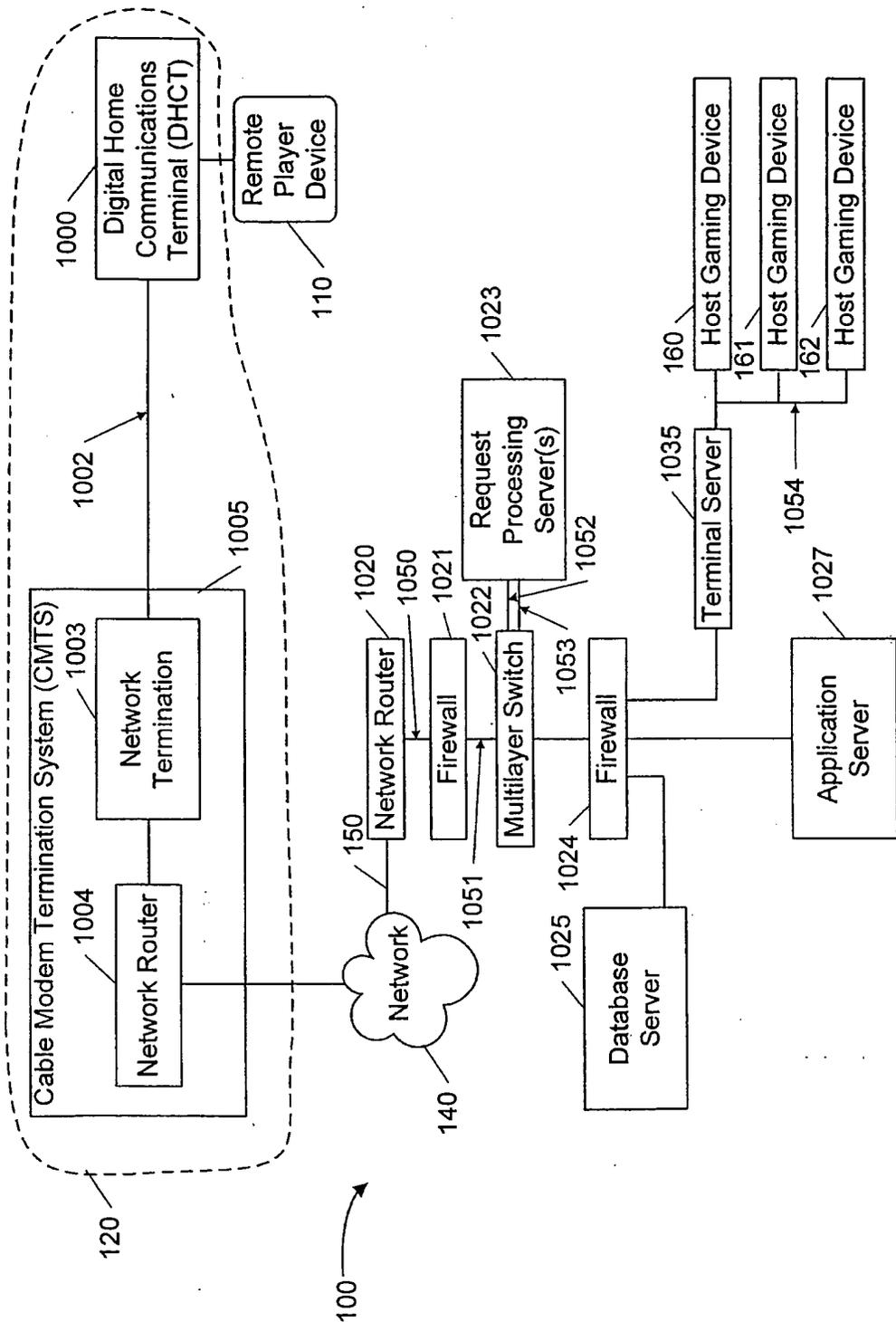
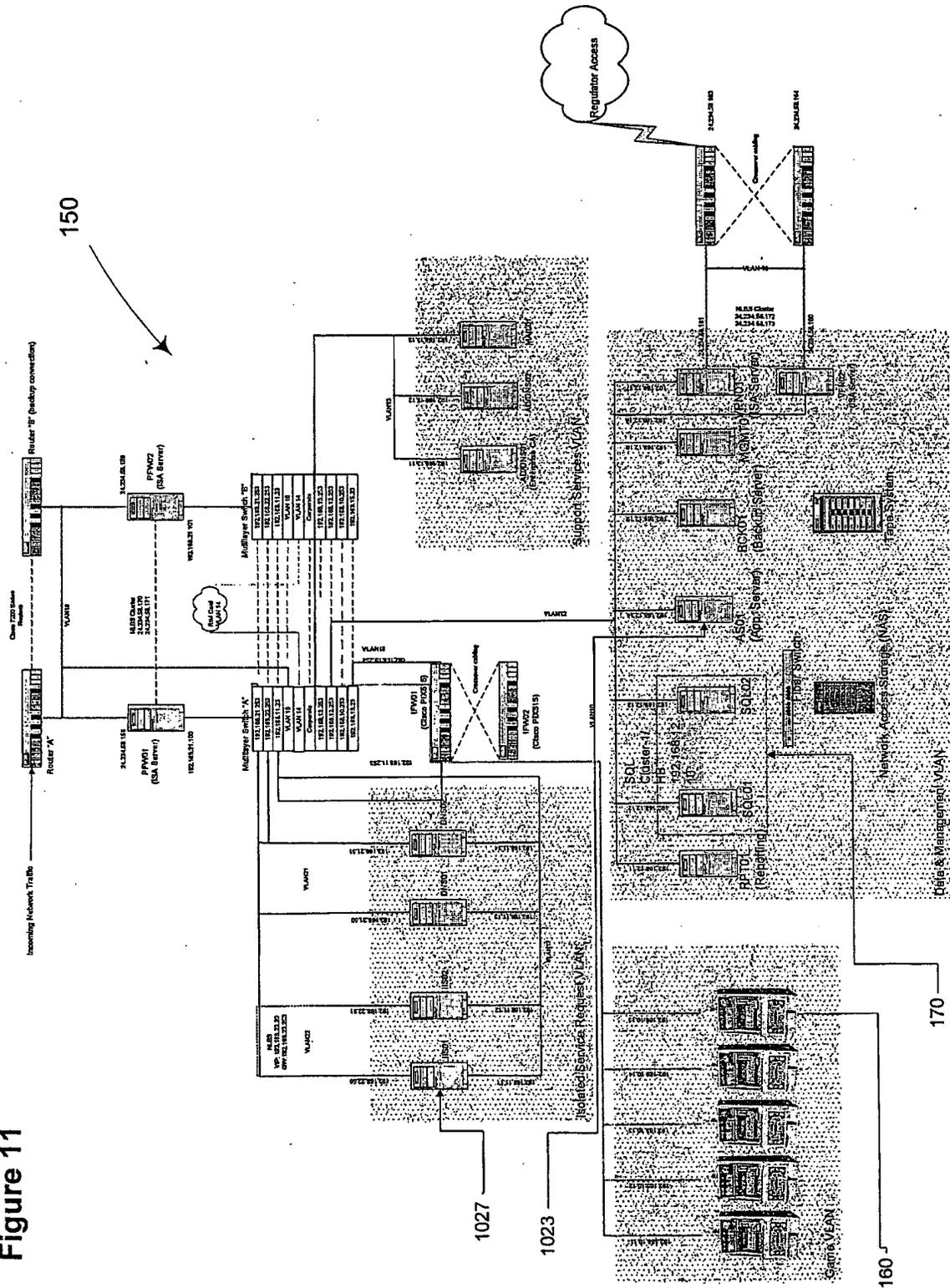


FIG. 10

Figure 11





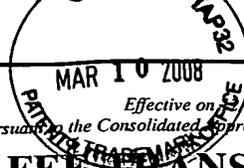
11w 3621

TRANSMITTAL FORM <i>(to be used for all correspondence after initial filing)</i>		Application Number	10/162,212
		Filing Date	June 5, 2002
		First Named Inventor	Xin WANG et al.
		Group Art Unit	3621
		Examiner Name	Augustin, Evens J.
Total Number of Pages in This Submission		Attorney Docket Number	111325-230300

ENCLOSURES <i>(check all that apply)</i>		
<input checked="" type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment / Reply <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input checked="" type="checkbox"/> Information Disclosure Statement <input type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Response to Missing Parts/Incomplete Application <input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____	<input type="checkbox"/> After Allowance Communication to Group <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input type="checkbox"/> Appeal Communication to Group (<i>Appeal Notice, Brief, Reply Brief</i>) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input checked="" type="checkbox"/> Other Enclosure(s) <i>(please identify below)</i> : 1. PTO Form 1449 2. One Box including 112 cited references
Remarks		<input checked="" type="checkbox"/> The Director is hereby authorized to charge any additional fees required or credit any overpayments to Deposit Account No. 19-2380 for the above identified docket number.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT	
Firm or Individual name	Marc S. Kaufman Registration No. 35,212 Nixon Peabody LLP 401 9th Street, N.W., Suite 900 Washington, D.C. 20004-2128
Signature	/Marc S. Kaufman, Reg. # 35,212/
Date	March 10, 2008

CERTIFICATE OF MAILING OR TRANSMISSION			
I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office (Fax No. (571) 273-8300) on the date shown below.			
Name <i>(Print/Type)</i>			
Signature		Date	



Effective on 08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL FOR FY 2008

Complete if Known

<input type="checkbox"/> Applicant claims small entity status. See 37 CFR 1.27		Application Number	10/162,212
TOTAL AMOUNT OF PAYMENT		Filing Date	June 5, 2002
\$180.00		First Named Inventor	Xin WANG et al.
		Examiner Name	Augustin, Evens J.
		Art Unit	3621
		Attorney Docket No.	111325-230300

METHOD OF PAYMENT (check all that apply)

Check Credit Card Money Order None Other (please identify): _____

Deposit Account Deposit Account Number: 19-2380 Deposit Account Name: Nixon Peabody LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

Charge fee(s) indicated below Charge fee(s) indicated below, except for the filing fee

Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-20238.

FEE CALCULATION

1. BASIC FILING, SEARCH AND EXAMINATION FEES

Application Type	FILING FEES		SEARCH FEES		EXAMINATION FEES		Fees Paid (\$)
	Fee (\$)	Small Entity Fee (\$)	Fee (\$)	Small Entity Fee (\$)	Fee (\$)	Small Entity Fee (\$)	
Utility	310	155	510	255	210	105	_____
Design	210	105	100	50	130	65	_____
Plant	210	105	310	155	160	80	_____
Reissue	310	155	510	255	620	310	_____
Provisional	210	105	N/A	N/A	N/A	N/A	_____

2. EXCESS CLAIM FEES

Fee Description	Fee (\$)	Small Entity Fee (\$)
Each claim over 20 (including Reissues)	50	25
Each independent claim over 3 (including Reissues)	210	105
Multiple dependent claims	370	185

Total Claims	Extra Claims	Fee (\$)	Fee Paid (\$)	Multiple Dependent Claims	Fee (\$)	Fee Paid (\$)
_____ - 20 or HP = _____	x _____	= _____	= _____	_____	_____	_____

HP = highest number of total claims paid for, if greater than 20

Indep. Claims	Extra Claims	Fee (\$)	Fee Paid (\$)
_____ - 3 or HP = _____	x _____	= _____	= _____

HP = highest number of independent claims paid for, if greater than 3

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$260 (\$130 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee (\$)	Fee Paid (\$)
_____ - 100 = _____	/ 50 = _____	(round up to a whole number)	x _____	= _____

4. OTHER FEE(S)

Description	Amount	Fees Paid (\$)
Non-English Specification,	\$130 fee (no small entity discount)	_____
Other: Information Disclosure Filing Fee		\$180.00

SUBMITTED BY

Signature	/Marc S. Kaufman, Reg. # 35,212/	Registration No. 35,212 (Attorney/Agent)	Telephone (202) 585-8000
Name (Print/Type)	Marc S. Kaufman		Date March 10, 2008

CERTIFICATE OF MAILING OR TRANSMISSION [35 CFR 1.8(a)]

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on _____.

Signature: _____
Name: _____

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)	Confirmation No.: 3700
Xin WANG et al.)	Group Art Unit: 3621
Application No.: 10/162,212)	Examiner: Augustin, Evens J.
Filed: June 5, 2002)	
For: RIGHTS OFFERING AND GRANTING)	Date: March 10, 2008
)	

INFORMATION DISCLOSURE STATEMENT

United States Patent and Trademark Office
Customer Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Dear Sir:

In accordance with the duty of disclosure as set forth in 37 C.F.R. § 1.56, Applicants hereby submit the following information in conformance with 37 C.F.R. §§ 1.97 and 1.98. Pursuant to 37 C.F.R. § 1.98(a)(2)(ii), copies of the cited U.S. patents (*i.e.*, Reference Cite Nos. 1–101) are not enclosed. Copies of the cited Foreign patents (*i.e.*, Reference Cite Nos. 102–173) are enclosed. Copies of the cited non-patent references (*i.e.*, Reference Cite Nos. 174–213) are enclosed. The references have been cited in recent oppositions in the European Patent Office relating to cases owned by assignee.

The Commissioner is hereby authorized to charge the **Deposit Account No. 19-2380** in the amount of **\$180.00** representing filing fees.

It is requested that the accompanying PTO/SB/08A be considered and made of record in the above-identified application. To assist the Examiner, the documents are listed on the attached form PTO/SB/08A. It is respectfully requested that an Examiner initialed copy of this form be returned to the undersigned.

The Commissioner is hereby authorized to charge any fees connected with this filing which may be required, or credit any overpayment to Deposit Account No. 19-2380.

03/11/2008 MAHMED1 00000109 192300 10162212
01 FC:1006 180.00 DA

Respectfully submitted,
NIXON PEABODY LLP

Date: March 10, 2008

By: /Marc S. Kaufman, Reg. # 35,212/
Marc S. Kaufman
Registration No. 35,212

NIXON PEABODY LLP
CUSTOMER NO.: 22204
401 9th Street, N.W., Suite 900
Washington, DC 20004
Tel: 202-585-8000
Fax: 202-585-8080



Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449A/PTO		<i>Complete if Known</i>	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>		Application Number	10/162,212
		Filing Date	June 5, 2002
		First Named Inventor	Xin Wang et al.
		Art Unit	3621
		Examiner Name	Augustin, Evens J.
Sheet 1	of 9	Attorney Docket Number	111325/230300

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	U.S. Patent Document Number - Kind Code ² (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	1	US 20010009026 A1	07-19-2001	Terao et al.	
	2	US 20010011276 A1	08-02-2001	Durst Jr. et al.	
	3	US 20010014206 A1	08-16-2001	Artigalas et al.	
	4	US 20010037467 A1	11-01-2001	O'Toole Jr. et al.	
	5	US 20010039659 A1	11-08-2001	Simmons et al.	
	6	US 20020001387 A1	01-03-2002	Dillon	
	7	US 20020035618 A1	03-21-2002	Mendez et al.	
	8	US 20020044658 A1	04-18-2002	Wasilewski et al.	
	9	US 20020056118 A1	05-09-2002	Hunter et al.	
	10	US 20020069282 A1	06-06-2002	Reisman	
	11	US 20020099948 A1	07-25-2002	Kocher et al.	
	12	US 20020127423 A1	09-12-2002	Kayanakis	
	13	US 20030097567 A1	05-22-2003	Terao et al.	
	14	US 20040052370 A1	03-18-2004	Katznelson	
	15	US 20040172552 A1	09-02-2004	Boyles et al.	
	16	US 4,159,468	06-26-1979	Barnes et al.	
	17	US 4,200,700	04-29-1980	Mäder	
	18	US 4,361,851	11-30-1982	Asip et al.	
	19	US 4,423,287	12-27-1983	Zeidler	
	20	US 4,429,385	01-31-1984	Cichelli et al.	
	21	US 4,621,321	11-04-1986	Boebert et al.	
	22	US 4,736,422	04-05-1988	Mason	
	23	US 4,740,890	04-26-1988	William	
	24	US 4,796,220	01-03-1989	Wolfe	
	25	US 4,816,655	03-28-1989	Musyck et al.	
	26	US 4,888,638	12-19-1989	Bohn	
	27	US 4,937,863	06-26-1990	Robert et al.	
	28	US 4,953,209	08-28-1990	Ryder et al.	
	29	US 4,977,594	12-11-1990	Shear	
	30	US 5,014,234	05-07-1991	Edwards	
	31	US 5,129,083	07-07-1992	Cutler et al.	
	32	US 5,138,712	08-11-1992	Corbin	
	33	US 5,174,641	12-29-1992	Lim	
	34	US 5,204,897	04-20-1993	Wyman	
	35	US 5,247,575	09-21-1993	Sprague et al.	
	36	US 5,260,999	11-09-1993	Wyman	
	37	US 5,276,444	01-04-1994	McNair	
	38	US 5,291,596	03-01-1994	Mita	
	39	US 5,293,422	03-08-1994	Loiacono	

Examiner Signature	Date Considered
--------------------	-----------------

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

10886567.1

Substitute for form 1449A/PTO		Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>		Application Number	10/162,212
		Filing Date	June 5, 2002
		First Named Inventor	Xin Wang et al.
		Art Unit	3621
		Examiner Name	Augustin, Evens J.
Sheet	2	of	9
		Attorney Docket Number	111325/230300

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	U.S. Patent Document Number - Kind Code ² (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	40	US 5,335,275	08-02-1994	Millar et al.	
	41	US 5,337,357	08-09-1994	Chou et al.	
	42	US 5,386,369	01-31-1995	Christiano	
	43	US 5,453,601	09-26-1995	Rosen	
	44	US 5,485,577	01-16-1996	Eyer et al.	
	45	US 5,504,816	04-02-1996	Hamilton et al.	
	46	US 5,530,235	06-25-1996	Stefik et al.	
	47	US 5,535,276	07-09-1996	Ganesan	
	48	US 5,557,678	09-17-1996	Ganesan	
	49	US 5,629,980	05-13-1997	Stefik et al.	
	50	US 5,636,346	06-03-1997	Saxe	
	51	US 5,638,443	06-10-1997	Stefik et al.	
	52	US 5,708,709	01-13-1998	Rose	
	53	US 5,715,403	02-03-1998	Stefik	
	54	US 5,745,879	04-28-1998	Wyman	
	55	US 5,764,807	06-09-1998	Pearlman et al.	
	56	US 5,765,152	06-09-1998	Erickson	
	57	US 5,787,172	07-28-1998	Arnold	
	58	US 5,790,677	08-04-1998	Fox et al.	
	59	US 5,812,664	09-22-1998	Bernobich et al.	
	60	US 5,825,876	10-20-1998	Peterson	
	61	US 5,825,879	10-20-1998	Davis	
	62	US 5,838,792	11-17-1998	Ganesan	
	63	US 5,848,154	12-08-1998	Nishio et al.	
	64	US 5,848,378	12-08-1998	Shelton et al.	
	65	US 5,850,433	12-15-1998	Van Oorschot et al.	
	66	US 5,915,019	06-22-1999	Ginter et al.	
	67	US 5,917,912	06-29-1999	Ginter et al.	
	68	US 5,933,498	08-03-1999	Schneck et al.	
	69	US 5,940,504	08-17-1999	Griswold	
	70	US 5,982,891	11-09-1999	Ginter et al.	
	71	US 5,987,134	11-16-1999	Shin et al.	
	72	US 5,999,624	12-07-1999	Hopkins	
	73	US 6,006,332	12-21-1999	Rabne et al.	
	74	US 6,020,882	02-01-2000	Kinghorn et al.	
	75	US 6,047,067	04-04-2000	Rosen	
	76	US 6,073,234	06-06-2000	Kigo et al.	
	77	US 6,091,777	07-18-2000	Guetz et al.	
	78	US 6,112,239	08-29-2000	Kenner et al.	

Examiner Signature	Date Considered
--------------------	-----------------

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

10886567.1

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449A/PTO		<i>Complete if Known</i>	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>		Application Number	10/162,212
		Filing Date	June 5, 2002
		First Named Inventor	Xin Wang et al.
		Art Unit	3621
		Examiner Name	Augustin, Evens J.
Sheet	3	of	9
		Attorney Docket Number	111325/230300

U.S. PATENT DOCUMENTS						
Examiner Initials*	Cite No. ¹	U.S. Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code ² (if known)				
	79	US 6,135,646		10-24-2000	Kahn et al.	
	80	US 6,141,754		10-31-2000	Choy	
	81	US 6,157,719		12-05-2000	Wasilewski et al.	
	82	US 6,169,976 B1		01-02-2001	Colosso	
	83	US 6,185,683 B1		02-06-2001	Ginter et al.	
	84	US 6,189,037 B1		02-13-2001	Adams et al.	
	85	US 6,189,146 B1		02-13-2001	Misra et al.	
	86	US 6,209,092 B1		03-27-2001	Linnartz	
	87	US 6,216,112 B1		04-10-2001	Fuller et al.	
	88	US 6,219,652 B1		04-17-2001	Carter et al.	
	89	US 6,236,971 B1		05-22-2001	Stefik et al.	
	90	US 6,307,939 B1		10-23-2001	Vigarie	
	91	US 6,353,888 B1		03-05-2002	Kakehi et al.	
	92	US 6,397,333 B1		05-28-2002	Söhne et al.	
	93	US 6,401,211 B1		06-04-2002	Brezak Jr. et al.	
	94	US 6,405,369 B1		06-11-2002	Tsuria	
	95	US 6,424,717 B1		07-23-2002	Pinder et al.	
	96	US 6,424,947 B1		07-23-2002	Tsuria et al.	
	97	US 6,487,659 B1		11-26-2002	Kigo et al.	
	98	US 6,516,052 B2		02-04-2003	Voudouris	
	99	US 6,516,413 B1		02-04-2003	Aratani et al.	
	100	US 6,523,745 B1		02-25-2003	Tamori	
	101	US 6,796,555 B1		09-28-2004	Blahut	

Examiner Signature		Date Considered	
--------------------	--	-----------------	--

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

10886567.1

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449A/PTO				<i>Complete if Known</i>	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Application Number	10/162,212
				Filing Date	June 5, 2002
				First Named Inventor	Xin Wang et al.
				Art Unit	3621
				Examiner Name	Augustin, Evens J.
Sheet	4	of	9	Attorney Docket Number	111325/230300

FOREIGN PATENT DOCUMENTS							
Examiner Initials ¹	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ²
		Country Code ¹	Number ⁴ Kind Code ⁵ <i>(if known)</i>				
	102	WO	83/04461 A1	12-22-1983	Western Electric Company, Inc.		
	103	WO	92/20022 A1	11-12-1992	Digital Equipment Corporation		
	104	WO	93/01550 A1	01-21-1993	Infologic Software, Inc.		
	105	WO	93/11480 A1	06-10-1993	Intergraph Corporation		
	106	WO	94/03003 A1	02-03-1994	Crest Industries, Inc.		
	107	WO	96/24092 A2	08-08-1996	Benson		
	108	WO	96/27155 A2	09-06-1996	Electronic Publishing Resources, Inc.		
	109	WO	97/25800 A1	07-17-1997	Mytec Technologies Inc.		
	110	WO	97/37492 A1	10-09-1997	Macrovision Corporation		
	111	WO	97/41661 A2	11-06-1997	Motorola Inc.		
	112	WO	97/43761 A2	11-20-1997	Intertrust Technologies Corp.		
	113	WO	98/09209 A1	03-05-1998	Intertrust Technologies Corp.		
	114	WO	98/10561 A1	03-12-1998	Telefonaktiebolaget LM Ericsson		
	115	WO	98/11690 A1	03-19-1998	Glover		
	116	WO	98/19431 A1	05-07-1998	Qualcomm Incorporated		
	117	WO	98/43426 A1	10-01-1998	Canal+Societe Anonyme		
	118	WO	98/45768 A1	10-15-1998	Northern Telecom Limited		
	119	WO	99/24928 A2	05-20-1999	Intertrust Technologies Corp.		
	120	WO	99/34553 A1	07-08-1999	V-One Corporation		
	121	WO	99/35782 A1	07-15-1999	Cryptography Research, Inc.		
	122	WO	99/48296 A1	09-23-1999	Intertrust Technologies Corporation		
	123	WO	99/60461 A1	11-25-1999	International Business Machines Corporation		
	124	WO	99/60750 A2	11-25-1999	Nokia Networks Oy		
	125	WO	00/04727 A2	01-27-2000	Koninklijke Philips Electronics N.V.		
	126	WO	00/05898 A2	02-03-2000	Optivision, Inc.		
	127	WO	00/59152 A2	10-05-2000	Microsoft Corporation		
	128	WO	00/72118 A1	11-30-2000	Compaq Computers Inc.		
	129	WO	00/73922 A2	12-07-2000	Entera, Inc.		
	130	WO	01/37209 A1	05-25-2001	Teralogic, Inc.		
	131	EP	0 067 556 B1	12-22-1982	Data General Corporation		
	132	EP	0 257 585 A2	03-02-1988	NEC Corporation		

Examiner Signature	Date Considered
-----------------------	--------------------

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

10886567.1

Substitute for form 1449A/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Application Number	10/162,212
				Filing Date	June 5, 2002
				First Named Inventor	Xin Wang et al.
				Art Unit	3621
				Examiner Name	Augustin, Evens J.
Sheet	5	of	9	Attorney Docket Number	111325/230300

FOREIGN PATENT DOCUMENTS							
Examiner Initials ¹	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ²
		Country Code ³	Number ⁴				
	133	EP 0 332 304	A2	09-13-1989	Digital Equipment Corporation		
	134	EP 0 393 806	A2	10-24-1990	TRW Inc.		
	135	EP 0 450 841	A2	10-09-1991	GTE Laboratories Incorporated		
	136	EP 0 529 261	A2	03-03-1993	International Business Machines Corporation		
	137	EP 0 613 073	A1	08-31-1994	International Computers Limited		
	138	EP 0 678 836	A1	10-25-1995	Tandem Computers Incorporated		
	139	EP 0 679 977	A1	11-02-1995	International Business Machines Incorporated		
	140	EP 0 715 243	A1	06-05-1996	Xerox Corporation		
	141	EP 0 715 244	A1	06-05-1996	Xerox Corporation		
	142	EP 0 715 245	A1	06-05-1996	Xerox Corporation		
	143	EP 0 731 404	A1	09-11-1996	International Business Machines Corporation		
	144	EP 0 763 936	A2	03-19-1997	LG Electronics Inc.		
	145	EP 0 818 748	A2	01-14-1998	Murakoshi, Hiromasa		
	146	EP 0 840 194	A2	05-06-1998	Matsushita Electric Industrial Co., Ltd.		
	147	EP 0 892 521	A2	01-20-1999	Hewlett-Packard Company		
	148	GB 1483282		08-17-1977	Compagnie Internationale Pour L'Informatique C11-Honeywell-Bull		
	149	GB 2236604	A	04-10-1991	Sun Microsystems Inc.		
	150	GB 2309364	A	07-23-1997	Northern Telecom Limited		
	151	GB 2316503	A	02-25-1998	ICL Personal Systems Oy		
	152	BR 9810967	A (Abstract only)	10-30-2001	Scientific Atlanta Inc.		
	153	EP 0 934 765	A1	08-11-1999	Canal+Societe Anonyme		
	154	EP 0 946 022	A2	09-29-1999	Nippon Telegraph and Telephone Corporation		
	155	EP 0 964 572	A1	12-15-1999	Canal+Societe Anonyme		
	156	EP 1 103 922	A2 (Abstract only)	05-30-2001	CIT Alcatel		
	157	GB 2022969	A	12-19-1979	Data Recall Limited		
	158	GB 2354102	A	03-14-2001	Barron McCann Limited		
	159	JP 11031130	A2 (Abstract only)	02-02-1999	Fuji Xerox Co. Ltd.		

Examiner Signature	Date Considered
--------------------	-----------------

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>			Complete if Known		
			Application Number	10/162,212	
			Filing Date	June 5, 2002	
			First Named Inventor	Xin Wang et al.	
			Art Unit	3621	
			Examiner Name	Augustin, Evens J.	
Sheet	6	of	9	Attorney Docket Number	111325/230300

FOREIGN PATENT DOCUMENTS							
Examiner Initials*	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ²
		Country Code ³	Number ⁴ Kind Code ⁵ <i>(if known)</i>				
	160	JP	11032037 A2 (Abstract only)	02-02-1999	Fuji Xerox Co. Ltd.		
	161	JP	11205306 A2 (Abstract only)	07-30-1999	Fuji Xerox Co. Ltd.		
	162	JP	11215121 A2 (Abstract only)	08-06-1999	Fuji Xerox Co. Ltd.		
	163	JP	2000215165 A2 (Abstract only)	08-04-2000	Nippon Telegraph and Telephone		
	164	JP	2005218143 A2 (Abstract only)	08-11-2005	Scientific Atlanta Inc.		
	165	JP	2005253109 A2 (Abstract only)	09-15-2005	Scientific Atlanta Inc.		
	166	JP	2006180562 A2 (Abstract only)	07-06-2006	Intarsia Software LLC; Mitsubishi Corp.		
	167	JP	5168039 A2 (Abstract only)	07-02-1993	Sony Corp.		
	168	WO	96/13814 A1	05-09-1996	Vazvan		
	169	WO	00/46994 A1	08-10-2000	Canal+Societe Anonyme		
	170	WO	00/62260 A1 (Abstract only)	10-19-2000	Swisscom Mobile AG		
	171	WO	01/03044 A1	01-11-2001	Transcast International, Inc.		
	172	WO	04/103843 (Abstract only)	12/02/2004	S2F Flexico		
	173	WO	04/34223 A2	04-22-2004	Legal IGaming, Inc.		

Examiner Signature		Date Considered	
-----------------------	--	--------------------	--

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

10886567.1

Substitute for form 1449A/PTO		Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>		Application Number	10/162,212
		Filing Date	June 5, 2002
		First Named Inventor	Xin Wang et al.
		Art Unit	3621
		Examiner Name	Augustin, Evens J.
Sheet	7	of	9
		Attorney Docket Number	111325/230300

OTHER PRIOR ART - NON PATENT LITERATURE DOCUMENTS			
Examiner Initials*	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ²
	174	BLAZE et al, "Divertible Protocols and Atomic Proxy Cryptography" 1998 Advances in Cryptography - Euro Crypt International Conference on the Theory and Application of Crypto Techniques, Springer Verlag, DE	
	175	BLAZE et al, "Atomic Proxy Cryptography" DRAFT (Online) (November 2, 1997) XP002239619 Retrieved from the Internet	
	176	NO AUTHOR, "Capability- and Object-Based Systems Concepts," Capability-Based Computer Systems, pp. 1-19 (no date)	
	177	COX, "Superdistribution" Wired Magazine (September 1994) XP002233405 URL: http://www.wired.com/wired/archive/2.09/superdis_pr.html&gt	
	178	DUNLOP et al, Telecommunications Engineering, pp. 346-352 (1984)	
	179	ELGAMAL, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory IT-31(4):469-472 (July 1985)	
	180	GHEORGHIU et al, "Authorization for Metacomputing Applications" (no date)	
	181	IANNELLA, ed., Open Digital Rights Language (ODRL), pp. 1-31 (November 21, 2000)	
	182	KAHLE, wais.concepts.txt, Wide Area Information Server Concepts, Thinking Machines Version 4, Draft, pp. 1-18 (November 3, 1989)	
	183	KAHN, "Deposit, Registration and Recordation in an Electronic Copyright Management System," Technical Report, Corporation for National Research Initiatives, Reston, Virginia (August 1992) URL: http://www.cni.org/docs/ima.ip-workshop/kahn.html	
	184	KAHN et al, "The Digital Library Project, Volume 1: The World of Knowbots (DRAFT), An Open Architecture for a Digital Library System and a Plan for its Development," Corporation for National Research Initiatives, pp. 1-48 (March 1988)	
	185	KOHL et al, Network Working Group Request for Comments: 1510, pp. 1-112 (September 1993)	
	186	LEE et al, CDMA Systems Engineering Handbook (1998) [excerpts but not all pages numbered]	
	187	MAMBO et al, "Protection of Data and Delegated Keys in Digital Distribution," Information Security and Privacy. Second Australian Conference, ACISP '97 Proceedings, pp. 271-282 (Sydney, NSW, Australia, 7-9 July 1997, 1997 Berlin, Germany, Springer-Verlag, Germany), XP008016393 ISBN: 3-540-63232-8	
	188	MAMBO et al, "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts," IEICE Trans. Fundamentals VOL. E80-A, NO. 1:54-63 (January 1997) XP00742245 ISSN: 0916-8508	
	189	Microsoft Word, Users Guide, Version 6.0, pp. 487-89, 549-55, 560-64, 572-75, 599-613, 616-31 (1993)	
	190	OJANPERÄ and PRASAD, eds., Wideband CDMA for Third Generation Mobile Communications (1998) [excerpts but not all pages numbered]	
	191	PERRITI, "Knowbots, Permissions Headers and Contract Law," Paper for the Conference on Technological Strategies for Protecting Intellectual Property in the Networked Multimedia Environment, pp. 1-22 (April 2-3, 1993 with revisions of April 30, 1993)	

Examiner Signature		Date Considered	
-----------------------	--	--------------------	--

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449A/PTO		<i>Complete if Known</i>	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>		Application Number	10/162,212
		Filing Date	June 5, 2002
		First Named Inventor	Xin Wang et al.
		Art Unit	3621
		Examiner Name	Augustin, Evens J.
		Attorney Docket Number	111325/230300
Sheet	8	of	9

OTHER PRIOR ART - NON PATENT LITERATURE DOCUMENTS			
Examiner Initials*	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ²
	192	RAGGETT, (Hewlett Packard), "HTML+(Hypertext markup language)," pp. 1-31 (12 July 1993) URL:http://citeseer.ist.psu.edu/correct/340709	
	193	SAMUELSON et al, "Intellectual Property Rights for Digital Library and Hypertext Publishing Systems: An Analysis of Xanadu," Hypertext '91 Proceedings, pp. 39-50 (December 1991)	
	194	NO AUTHOR, "Softlock Services Introduces... Softlock Services" Press Release (January 28, 1994)	
	195	NO AUTHOR, "Appendix III - Compatibility with HTML," NO TITLE, pp. 30-31 (no date)	
	196	NO EDITOR, NO TITLE, Dictionary pages, pp. 469-72, 593-94 (no date)	
	197	BENOIT, Digital Television MPEG-1, MPEG-2 and Principles of the DVB System, pp. 75-80, 116-121 (no date)	
	198	BENOIT, Digital Television MPEG-1, MPEG-2 and Principles of the DVB System, 2 nd edition, pp. 74-80 (no date)	
	199	AH Digital Audio and Video Series, "DTV Receivers and Measurements," Understanding Digital Terrestrial Broadcasting, pp. 159-64 (no date)	
	200	O'DRISCOLL, The Essential Guide to Digital Set-Top Boxes and Interactive TV, pp. 6-24 (no date)	
	201	IUS MENTIS, "The ElGamal Public Key System," pp. 1-2 (October 1, 2005) online at http://www.iusmentis.com/technology/encryption/elgamal/	
	202	SCHNEIER, "Crypto Bibliography," Index of Crypto Papers Available Online, pp. 1-2 (online) (no date)	
	203	NO AUTHOR, NO TITLE, pp. 344-55 (no date)	
	204	NO AUTHOR, "Part Four Networks," NO TITLE, pp. 639-714 (no date)	
	205	Microsoft Word User's Guide, pp. 773-74, 315-16, 487-89, 561-64, 744, 624-33 (1993)	
	206	NO AUTHOR, "What is the ElGamal Cryptosystem," p. 1 (November 27, 2006) online at http://www.x5.net/faqs/crypto/q29.html	
	207	JOHNSON et al., "A Secure Distributed Capability Based System," ACM, pp. 392-402 (1985)	
	208	Wikipedia, "El Gamal Encryption," pp.1-3 (last modified November 2, 2006) online at http://en.wikipedia.org/wiki/ElGamal_encryption	
	209	BLAZE, "Atomic Proxy Cryptography," p. 1 Abstract (October 20, 1998)	
	210	BLAZE, "Matt Blaze's Technical Papers," pp. 1-6 (last updated August 6, 2006)]	
	211	Online Search Results for "inverted file", "inverted index" from www.techweb.com , www.cryer.co.uk , computing-dictionary.thefreedictionary.com , www.nist.gov , en.wikipedia.org , www.cni.org , www.tiscali.co.uk (July 15-16, 2006)	
	212	Corporation for National Research Initiatives, "Digital Object Architecture Project", http://www.nnri.reston.va.us/doa.html (updated 28 Nov 2006)	

Examiner Signature	Date Considered
--------------------	-----------------

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				<i>Complete if Known</i>		
				Application Number	10/162,212	
Sheet		9	of	9	Filing Date	June 5, 2002
					First Named Inventor	Xin Wang et al.
					Art Unit	3621
					Examiner Name	Augustin, Evens J.
					Attorney Docket Number	111325/230300

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS			
Examiner Initials ¹	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ²
	213	STEFIK, Summary and Analysis of A13 (Kahn, Robert E and Vinton G Cerf, "The Digital Library Project, Volume 1: The World of Knowbots (DRAFT), An Open Architecture for a Digital Library System and a Plan for its Development," Corporation for National Research Initiatives (March 1988)), pp. 1-25 (May 30, 2007)	

Examiner Signature		Date Considered	
--------------------	--	-----------------	--

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

10886567.1



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

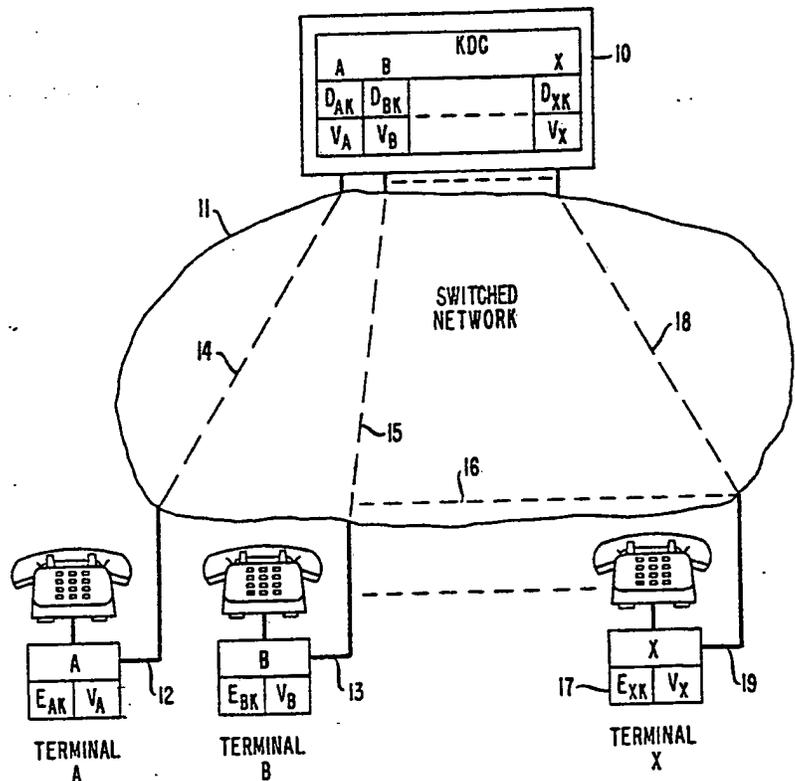
<p>(51) International Patent Classification³ : H04L 9/00</p>	<p>A1</p>	<p>(11) International Publication Number: WO 83/ 04461 (43) International Publication Date: 22 December 1983 (22.12.83)</p>
<p>(21) International Application Number: PCT/US83/00030 (22) International Filing Date: 11 January 1983 (11.01.83) (31) Priority Application Number: 386,805 (32) Priority Date: 9 June 1982 (09.06.82) (33) Priority Country: US (71) Applicant: WESTERN ELECTRIC COMPANY, INC. [US/US]; 222 Broadway, New York, NY 10038 (US). (72) Inventors: EVERHART, Joseph, Robert ; P.O. Box 228, 3 Old Mill Road, Holmdel, NJ 07733 (US). OSBORN, Jeffrey, George ; 242 Madison Gardens, Old Bridge, NJ 08857 (US). (74) Agents: HIRSCH, A., E., Jr. et al.; Post Office Box 901, Princeton, NJ 08540 (US).</p>		<p>(81) Designated States: AT (European patent), AU, BE (European patent), CH (European patent), DE (European patent), FR (European patent), GB (European patent), JP, LU (European patent), NL (European patent), SE (European patent). Published <i>With international search report.</i></p>

(54) Title: **ENCRYPTION SYSTEM KEY DISTRIBUTION METHOD AND APPARATUS**

(57) Abstract

Encryption systems typically rely on the distribution of cipher keys between terminals for scrambling and unscrambling transmitted messages. Elaborate security precautions are necessary to protect the cipher keys since a compromise of the key could result in a compromise of the transmission. There is disclosed a key distribution method and apparatus which uses a channel (14, 15, 18) from identified terminals (A, B, X) to a central key distribution center (KDC) for the establishment, on a one-session basis, of the key which is to be used for the next session between those terminals. The key establishing link (16) is itself encoded using a cipher key which changes after each usage. Provision is made to verify, for each new connection, that a compromise has not priorly occurred.

KDC CONFIGURATION



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	LI	Liechtenstein
AU	Australia	LK	Sri Lanka
BE	Belgium	LU	Luxembourg
BR	Brazil	MC	Monaco
CF	Central African Republic	MG	Madagascar
CG	Congo	MR	Mauritania
CH	Switzerland	MW	Malawi
CM	Cameroon	NL	Netherlands
DE	Germany, Federal Republic of	NO	Norway
DK	Denmark	RO	Romania
FI	Finland	SE	Sweden
FR	France	SN	Senegal
GA	Gabon	SL	Soviet Union
GB	United Kingdom	TD	Chad
HU	Hungary	TG	Togo
JP	Japan	US	United States of America
KP	Democratic People's Republic of Korea		

- 1 -

ENCRYPTION SYSTEM KEY DISTRIBUTION
METHOD AND APPARATUS

Background of the Invention

5 This invention relates to the establishment and distribution of cipher keys in a cryptographic system.

Cryptographic systems are now gaining favor, both for voice as well as data transmission. In such systems it is typically necessary that the parties to a particular
10 transmission each have cryptographic keys to encrypt and decrypt the cipher transmissions. It follows that a compromise to a cryptographic key will in turn reduce the security of subsequent transmissions involving that key. Thus, great precautions must be taken to distribute the
15 cryptographic keys among the system users. Such distribution, for example, using secure couriers to manually update the keys may be possible when the community of users is priorly known but becomes increasingly more difficult when either the number of parties is large or
20 parties who seldom communicate with each other wish to do so. The responsibility for keeping the cryptographic key secure after distribution rests with each user and the longer the key remains effective the greater the risk of it becoming compromised.

25 Thus, from a practical point of view it is desirable to have the cryptographic key effective for a single session, requiring a new key for each new session. When couriers are used, however, this becomes costly and time consuming, especially when a party wishes to place
30 many secure calls or have many secure sessions.

Attempts have been made to electronically distribute cryptographic keys between users from a key distribution center. One such example is shown in Rosenblum Patent No. 4,182,933, issued January 8, 1980.
35 While such attempts have found some degree of success they all suffer from the problem that they are subject to

- 2 -

compromise because they usually rely on the security of the transmission media between the key distribution center and the terminal for the distribution of session key information. Thus, an intruder need only compromise the key distribution channel to obtain subsequent session keys. Elaborate systems have sometimes been established to detect such a compromise, all of which are either costly or minimally effective.

Another problem with key distribution centers is that the center can derive the information used to decrypt the secure data exchange between users and thus could theoretically monitor the secure session transmission.

Summary of the Invention

We have solved the above-identified problems by arranging a key distribution center (KDC) which communicates over a channel with the individual terminals. The channel, or data link, can be a dial-up telephone line, a packet-switched data network, dedicated lines, or other communications channel types, over which secure communication is possible. The terminals operate in conjunction with the KDC to establish a session key for secure transmission between two or more terminals. The session key at a terminal is constructed from information generated at that terminal in conjunction with information communicated from the KDC and is known fully only to the terminals involved in the session and not to the KDC. Thus, when two terminals have established a session key, they may securely communicate with each other for the duration of that session.

At the conclusions of the secure data exchange, the session keys should be destroyed, and when either station wishes to establish additional secure communication either between themselves or to other stations, a new session key will be established in cooperation with the KDC.

Both the terminal-KDC channel and the KDC-terminal channel, as mentioned above, are secure links in

SUBSTITUTE SHEET



- 3 -

that they are protected by cryptographic key information which is unique to each terminal and to the KDC on a one-call-only basis. Accordingly, whenever a connection is established between a terminal and the KDC, each has

5 information previously stored, referred to as terminal-unique key information, and this priorly stored information is used to establish both new KDC-terminal link keys, referred to as call-setup key information, and new session

10 keys, the terminal and the KDC each modify their respective terminal-unique key information so that on a next call between the KDC and the same terminal, this new key information must be used in order to establish a secure communication path. The precise manner in which this

15 happens will be discussed hereinafter. In this manner, an intruder on the key distribution between a terminal and the KDC must be adding and substituting information on the channel from the beginning and must stay on the channel throughout several calls, since once the intruder leaves it

20 is possible to detect, at least by hindsight, that a compromise has occurred. This is a result of the fact that the intruder is substituting random information that may be monitored.

One aspect of our system is that an intruder, in

25 order to obtain useful information exchanged between two valid users of the system, must gain the terminal-unique information that is stored at the terminal, and he must also gain the terminal-unique information that is stored in the key distribution center for that specific terminal.

30 The intruder then, on the very next key exchange involving that terminal and the key distributing center, must actively participate, i.e., substitute his own generated key information on that channel. Then the intruder must also substitute information on the channel between the two

35 communicating terminals, and also must continue the above substitutions on the channels for an indefinite period of time or risk detection.

SUBSTITUTE SHEET

Petitioner Apple Inc. - Ex. 1025, p. 5602



- 4 -

Brief Description of the Drawing

These attributes of our invention, together with the operation and utilization of the invention in a specific embodiment, will be more fully apparent from the illustrative embodiment shown in conjunction with the drawing which:

FIG. 1 shows an overall system using a KDC and several terminals;

FIG. 2 shows an implementation of the initial establishment of information in both the KDC and the terminal within a secure area;

FIGS. 3 and 4 show a flow chart detailing what occurs within each terminal;

FIG. 5 shows a flow chart detailing what occurs within the KDC;

FIGS. 6-19 show, in sequence, an implementation of the establishment of key information and control data within each terminal; and

FIGS. 21-28 show, in sequence, an implementation of the establishment of key information and control data within the KDC. In this system we have a variety of terminals.

General Description

FIG. 1 shows a number of terminals, A, B and X, connectable to each other and to KDC 10 via some transport network (e.g., public switched network). These terminals should be able to set up a secure channel between themselves in order to exchange secure information. In this process they must both communicate with the KDC. The transmission line 12 from terminal A is connected through link 16 to transmission line 13 to initiate a secure call to terminal B. Once the users decide to initiate a secure data exchange, each terminal sets up a transmission line, such as link 14 for terminal A, to the KDC.

An exchange of information will then occur from terminal A to the KDC and from terminal B to the KDC. Once the KDC has received both of these messages, it will

SUBSTITUTE SHEET

Petitioner Apple Inc. - Ex. 1025, p. 3603



- 5 -

formulate two distinct messages that will be sent respectively to terminal A via link 14 and to terminal B via link 15. These individual messages will contain session key information, as well as other pertinent information described below. This session key information has originated at terminal A and at terminal B and is exchanged through the KDC. Once the exchange has taken place between the two terminals and the KDC, link 14, which is the key distribution link between terminal A and the KDC, is then taken down, and key distribution link 15 between the KDC and terminal B is taken down. Link 16, which is the session link between terminals A and B, is re-established. Further key information is exchanged based on the prior partial exchanges so as to derive independently at both terminals the session key, and finally using that session key information, data (i.e., digital data or digital voice) can be transmitted in secure fashion on data link 16.

Since further session information was derived between terminals A and B independent of the KDC, a malicious operator of the KDC cannot derive the key information need to decrypt the secure messages sent between terminals A and B without actively substituting information on the session channel.

Also, at this point, as will be seen, contained within the messages that were sent between the KDC and the terminals was new terminal-unique key information to secure the next key distribution between the terminals and the KDC. This new information is independent of the previous information and therefore is unique to it.

Detailed Description

Turning now to FIG. 2 the initial setup between the terminal and the KDC must be made in an authentic manner such that the information transported to the terminals from the KDC is not modified. One implementation is where the transport is made within a secured area, such as secured area 23. Since subsequent communications

SUBSTITUTE SHEET



- 6 -

between the KDC and each terminal depend upon the prior communication, it is important that at some period in time they both contain the proper information for start-up, and ideally this is done in the secured area so that there can
5 be no breach of security.

On the initial system setup (based on the secured area implementation shown in FIG. 2) the terminals are brought within the secured area 23, and the KDC can generate terminal-unique key pairs for each terminal. The
10 exact function of these key pairs will be described later. The KDC will generate a terminal-unique decryption key for each terminal and the corresponding encryption key. This encryption key must be placed in the terminal-unique key storage for each terminal with the corresponding decryption
15 key stored in the terminal-unique key storage at the KDC under the address of that terminal. In addition, a random number, U_a for terminal A, unique to each terminal is stored in the verification information storage at the KDC also at the address of this terminal. This same random
20 number must be loaded and stored in the verification information storage in the terminals and will be used for a verification check on the first call setup to the KDC.

FIGS. 3 and 4 are flow charts representing the action that occurs within a terminal, for example,
25 terminal A.

FIG. 5 is a flow chart representing what actions occur within the key distribution center.

The discussion which will follow is a discussion with respect to a time sequence between the terminal and
30 the KDC to illustrate both how terminal-unique keys are updated, and how call-setup and session keys are distributed. This discussion will occur with respect to FIGS. 6 through 28. FIGS. 6 through 19 show the apparatus within the terminal and show on a step-by-step basis how
35 the call-setup keys and the session keys are established. FIGS. 20 through 28 show the apparatus within the KDC, each figure showing a specific operational aspect of the

SUBSTITUTE SHEET

- 7 -

establishment of the keys.

Turning now to FIG. 6, we will discuss the specific apparatus used in the terminals. The actual generation of the numbers will be discussed hereinafter.

5 Apparatus 72 is a random number generator which is a device or algorithm that produces bits (zeros and ones) that are equally likely to occur. This generation may be based upon a noisy diode and any number of algorithms can be used to attain statistically independent output of 0's and 1's.
10 The more equally likely these random number generators are, i.e., the more random this function is, the higher the security level will be. The output of the random number generator is a serial stream of zeroes and ones where the correlation between one or a group of bits is zero. The
15 bidirectional asymmetric key generator, apparatus 73, takes as input a random number from random number generator 72 and will compute an encryption key and the matching decryption key such that the encryption key cannot be derived from the decryption key and vice versa. The
20 generation of these keys as an example could be done in accordance with the RSA algorithm, as described by Rivest, Shamir, and Adleman in a paper entitled, "A Method for Obtaining Digital Signatures and Public Key Crypto Systems," which publication is hereby incorporated by
25 reference, which appeared in CACM, Vol. 21, No. 2, February, 1978, on pages 120-126.

Apparatus 74 implements a bidirectional asymmetric cryptographic algorithm (e.g., the RSA algorithm) that is, a cryptographic algorithm based on two
30 distinct keys where the encryption key cannot be derived from the decryption key and vice versa. Apparatus 74 has two inputs (I and K) and one output (O). The input I is the bits to be encrypted or decrypted. The input K is the key, either encryption or decryption (the RSA algorithm
35 performs the same function regardless of encryption or decryption). The output will be the inputted bits encrypted or decrypted with the supplied key. This

SUBSTITUTE SHEET



- 8 -

algorithm is also described in the aforementioned paper. Functionally, apparatus 75 is the embodiment of two functions f and g such that: given $f(R, P)$ and P , one cannot determine R ; $g(R1, f(R2, P), P) = g(R2, f(R1, P), P)$; and given $f(R1, P)$, $f(R2, P)$, and P one cannot determine $R1$, $R2$, or $g(R1, f(R2, P), P)$.

Apparatus 75 performs the above functions via, for example, the Diffie-Hellman algorithm, which is described in a paper by Diffie and Hellman entitled "New Directions in Cryptography," published by the IEEE Transactions on Information Theory, Vol. IP-22, November, 1976, on pages 644-655, which is hereby incorporated by reference. The input to this algorithm is a base Y , a modulus Q and an exponent EXP . The output is Y raised to the EXP power modulus the Q . The functions f and g are the same as discussed above in this example.

The storage requirements are depicted by registers 71, 70 and 76. These are the semi-permanent register 71 which contains both the verification information Va and the terminal-unique key information Eak used to encrypt messages to the KDC. Temporary register 70 can be in any state initially and is used during the interaction with the KDC on a secure call setup. The address register permanently contains the address (i.e., a public piece of information that uniquely identifies A to the KDC) of the terminal (terminal A in this case) where it is located. During a secure session (or call) setup, the address register will also contain the address of the terminal which is being called. The registers containing verification information and encryption and decryption information may vary in size depending upon the specific algorithm used but in this example should be on the order of 1,000 bits each. Information pertaining to the symmetric session key and the random number should be on the order of 100 bits, and the address information will be dependent upon a terminal numbering plan both unique and known to the KDC. For example, it could be the telephone

SUBSTITUTE SHEET

- 9 -

number of the specific terminal or it could be the serial number of the terminal.

Turning to FIG. 20, we will now discuss the working of the modules within the key distribution unit.

5 The address register at the KDC, register 200, performs the same function as the address register at the terminal. The RSA function at the KDC, apparatus 210, performs the same function as the RSA function at the terminal, as previously described. The random number generator, apparatus 211,
10 performs the same function as the random number generator at the terminal previously mentioned. The generator of the encryption and decryption keys apparatus 212 has the same function as described previously in the terminal. Apparatus 213 is a generator of the parameters used as
15 inputs to the apparatus 75 described previously. For this particular example these parameters are the base and modulus for the Diffie-Hellman algorithm. It requires as input the output of the random number generator, apparatus 211. The method of generation is described in
20 the aforementioned paper by Diffie.

There is a semi-permanent storage at the KDC, registers 214 and 216, which stores verification information V_a and terminal-unique decryption key information D_{ak} between calls. Semi-permanent
25 registers 215 and 217 are used to store information during the call setup progress. These registers have the same functions as described previously for the terminal.

System Operation

The operation of the system will now be explained
30 beginning with FIG. 3. Initially the key management equipment in the terminal will be in the wait state until a request is received from the terminal controller processor to initiate a secure call. At this point, as discussed, there is stored in the terminal the terminal-unique
35 encryption key that will be used to encrypt information that is sent to the KDC. Also stored is the verification information. These two pieces of information were stored

SUBSTITUTE SHEET



- 10 -

from the last call (or from the initial setup) that was made by this terminal. This is shown in FIG. 6 as Va and Eak.

Once a request is received to initiate a secure call, the address of the called party must be given to the key management equipment via the controller processor. This is seen in FIG. 3, box 31. At this point, there are generated new call-setup keys. This is shown in box 32 and in FIG. 7 as Eka and Dka. In box 33 there is shown the generation of partial session keys that will be used to encrypt data on the link from terminal B to terminal A. This is shown in FIG. 8 as Eba and Dba.

At this point, the verification information is updated using the keys that were just generated. The update function is specified as follows:

$$\text{Val}' = f(\text{Val}, \text{E1}) \text{ and } \text{Va2}' = f(\text{Va2}, \text{E2})$$

where ' denotes updated and $\text{ValVa2} = \text{Va}$. Va is the stored verification information and the E's are the just-generated encryption keys. The properties of f are as follows:

- (1) for every V, E1, E2: $f(V, E1) \neq f(V, E2)$ where $E1 \neq E2$;
- (2) for every V1, V2, E: $f(V1, E) \neq f(V2, E)$ where $V1 \neq V2$;
- (3) given V and $V' \neq f(V, E)$ it is difficult to determine E; and
- (4) in the case where E is an asymmetric encryption key, D cannot be determined from E.

For this example, $\text{Va}' = \text{Val}'|\text{Va2}'$ where $\text{Va} = \text{Val}|\text{Va2}$, Val' is equal to Val encrypted with Eka, and Va2' is equal to Va2 encrypted with Eba. This update process is depicted in FIG. 9. The first half of the verification information Val is read from storage and provided as an input to the RSA algorithm. The key that is used to encrypt this information is the call-setup key, Eka, that was just generated. This becomes Val' and overwrites Val as seen in

SUBSTITUTE SHEET



- 11 -

FIG. 10. Next, the second half of the verification information Va2 is encrypted using Eba just generated. The result Va2' overwrites Va2 in the storage register. This is shown in FIG. 3, box 34, and in summary, the updated verification information Va" is the verification information stored from the previous call, or given to the terminal on the initial setup from the KDC, where half is encrypted using the encryption part of the partial session key generated on this call and the other half is encrypted using the call-setup key for that call.

At this point, as shown in box 36, FIG. 3, and in FIG. 11, the message can be formatted to the KDC. The contents of this message are the encryption parts of the two keys that were just generated. Both the partial session key to be established between terminal A and B, Eba, and the new call-setup key Eka are encrypted using the terminal-unique encryption key Eak stored from the previous call from the KDC to the terminal or given to the terminal on the initial setup. At this point, the information that can be destroyed from the terminal is the terminal-unique encryption key, Eak, stored at the terminal from the previous call, and both the call-setup encryption key, Eka, and the partial session encryption key, Eba, that were generated by the terminal. The encrypted message is then appended to the address, A, of the originating terminal followed by the address, B, of the called terminal. This message is now sent to the KDC.

The terminal now will enter a wait state waiting for the information to be received from the KDC. This is depicted in box 37 of FIG. 3.

As shown in FIG. 5, the KDC will be in a wait state until a message is received from terminal A. This is shown in FIG. 5, box 50. Once the message is received, the KDC reads the address information within the message into the address register which gives it the index of the decryption key that must be used to decrypt the message. The KDC has in its storage from the previous call the

SUBSTITUTE SHEET



- 12 -

matching verification information for each terminal and the terminal-unique decryption key for each terminal. This is depicted in FIG. 20, boxes 214 and 216.

5 The message from terminal A is decrypted using the terminal-unique decryption key corresponding to that terminal, Dak. The keys, both the new call setup key Eka and the partial session key Eba (to be distributed to terminal B) is temporarily stored in the KDC memory as depicted in FIG. 21.

10 At this point, as shown in FIG. 22, the KDC can update its verification information in the exact same manner as the terminal. This is done by encrypting each half of the stored verification information Va with the received session key information Eba and the received
15 call-setup key information Eka, shown in FIG. 23. This produces the update verification information Va".

The key distribution center, as shown in FIG. 24, will now generate a bidirectional asymmetric encryption/decryption key pair, Eak', Dak'. The primes
20 denote updated information. Eak' will be distributed to terminal A to be used on the next call setup to the key distribution center. The decryption key Dak' overwrites the decryption key Dak that was stored from the previous call.

25 Two other pieces of information are also generated at this time. These are the parameters that will be used by the terminals to create symmetric session keys; in this case they are the parameters of the Diffie-Hellman algorithm. One is the base Y and the other is the
30 modulus Q as previously described. Functionally, the amount of information that is generated at the KDC and sent to each terminal may vary depending upon the precise algorithm. This information is stored in temporary storage and will be used as part of the message sent back to both
35 terminal A and terminal B. This generation process, is depicted in FIG. 25 and refers to the flow chart box 55, FIG. 5. By this point, as shown in FIG. 26, the KDC must

SUBSTITUTE SHEET

Petitioner Apple Inc. - Ex. 1025, p. 5611.

**BUREAU
OMPI**

- 13 -

have received a message from terminal B in order to complete the call to terminal A. If not, the KDC process for terminal A must wait until the process for terminal B has reached this point. This is so it can give terminal A
5 the partial session key information Eab generated at terminal B and also to be able to give terminal B the partial session key Eba generated at terminal A. Coordination between the processes must take place so that the same parameters generated by one process overwrites the
10 parameters generated by the other process. This insures that the parameters sent to the terminals for the purpose of generating symmetric session keys are the same.

Once the internal exchange is made between the A registers and the B registers to coordinate the information
15 inside the key distribution center, the messages can now be formatted for the terminals. This is shown in FIG. 27. The message to terminal A will consist of the new terminal-unique key information Eak' that will be used on a subsequent call to the KDC. It will also consist of the
20 partial session key information Eab which it received from terminal B. It will also consist of the verification information Va^n or a known reduction of Va^n in terms of the number of bits. It will also consist of the base Y and the modulus Q of the Diffie-Hellman algorithm. These five
25 pieces of information will be encrypted using the call-setup key Eka received in the message from terminal A. The KDC destroys Eka, Eba, Eak', Y, and Q corresponding to terminal A and destroys Ekb, Eab, Ebk', Y, and Q corresponding to terminal B. The KDC will then send this
30 output message back to terminal A. An analogous encrypted message is sent from the KDC to terminal B. At this point the KDC is finished with its processing.

FIG. 28 shows the configuration of the KDC after the call to terminal A has been dropped. The KDC has
35 updated verification information Va^n and updated terminal-unique decrypt key information Dak' which will be used on a subsequent call between terminal A and the KDC.

SUBSTITUTE SHEET

- 14 -

Referring back to the flow chart, FIG. 3, for terminal A, the key management equipment at the terminal has been in a wait state while the KDC has been functioning. FIG. 12 shows the key information stored at the terminal during this wait state. It is the updated verification V_a information and both decrypt keys D_{ka} and D_{ba} corresponding to the previously generated encryption keys.

FIG. 13 shows how the information received from the KDC is used in accordance with the box 38, FIG. 3. The call-setup decryption key D_{ka} is used to decrypt the message received from the KDC. The five values (previously discussed) sent from the KDC are now used in the following way. The first piece of information is the new distribution key E_{ak}' that is stored in the semi-permanent register 71 and will be used on a following call made from this terminal to the KDC. It is the updated terminal-unique encryption key. The second piece of information is the partial session key E_{ab} which was generated at B and sent through the KDC to terminal A. The third piece of information is the updated verification information V_a , which can now be compared with the verification information stored at terminal A. The fourth and fifth pieces of information are the parameters to the Diffie-Hellman algorithm, the base Y and the modulus Q , which terminal A stores in temporary storage.

Referring to FIG. 4, box 40, at this point the terminal will compare the verification information it received from the KDC and either the verification information which is presently stored or some known reduction of that verification information - FIG. 14. If this matches, then the process will continue as normal. If this does not match, an alarm could be given to the terminal controller processor of a potential intruder threat on a previous call.

Assuming a success of the compared verification, the terminal can now take down the channel to the KDC and

SUBSTITUTE SHEET

- 15 -

establish a channel to terminal B, if not already established. At this point, terminal A and terminal B can communicate data securely using the asymmetric session keys Eab and Eba. If a symmetric session key is needed, the following steps can be taken. The calculation of the message to be sent to terminal B is shown in FIG. 15. First, the base Y and modulus Q of the Diffie-Hellman algorithm are used along with a random number Ra generated by the random number generator 72. These inputs are given to the Diffie-Hellman algorithm 75 and the output is then an input to the RSA function 73. The random number Ra is also stored in temporary storage. Eab is used as the key to the RSA function 73. At this point the session key information Eab received from terminal B and the base number Y may be destroyed. The output of the RSA algorithm is sent to terminal B.

Terminal A' key management equipment will now enter a wait state shown in FIG. 4, box 44, waiting for a message to be returned from terminal B. The idle state is depicted in FIG. 16 and in storage is the decrypt session key Dab which terminal A generated, the modulus Q of the Diffie-Hellman algorithm generated by the KDC and the random Ra number that was generated by terminal A.

As shown in FIG. 17, upon receipt of the message from terminal B, terminal A will decrypt the message using its decryption key Dba stored from the initial generation of the partial session key. Dba can now be destroyed. The output of this will be fed into the Diffie-Hellman algorithm as the base. The exponent will be the random number Ra which was priorly generated and the modulus Q is also input into the algorithm. The output of the Diffie-Hellman algorithm will be symmetric session key information which will equal the session key information that terminal B has calculated. Q and Ra can now be destroyed.

At this point, terminals A and B have established symmetric session key information between themselves that is not derivable by the KDC. This key information may be

SUBSTITUTE SHEET

- 16 -

used in a symmetric key algorithm like the Data Encryption Standard (DES) to encrypt data. What is stored now in the terminal until the next request for a secure session (or call), as shown in FIG. 18, is the updated verification information Va" and the terminal-unique key Eak' which it received from the KDC to be used to encrypt the next message to the KDC.

It should be noted that the actual generation of the desired data at the terminal and at the KDC is operative under control of a computer processor and is programmed in accordance with the flow charts shown in FIGS. 3-5 to perform the sequence of data transfers detailed herein. Such a processor, while not shown, can be any one of several well-known microprocessors, such as for example, the Intel 8086 microprocessor, working in conjunction with the terminal and KDC apparatus shown and detailed herein above.

It should also be noted that one skilled in the art could use different encryption algorithms and different equipments to achieve the same results disclosed herein without departing from the spirit and scope of our invention.

SUBSTITUTE SHEET

- 17 -

Claims

1. A key distribution method for communicating cipher keys between two terminals via a key distribution center, KDC, said method comprising
- 5 establishing between any one terminal and said key distribution center a terminal-unique cipher key, cooperating between said KDC and said one terminal on a subsequent connection between said KDC and said one terminal to establish a session key for use by said one
- 10 terminal in a subsequent secure transmission between said one terminal and a second terminal, and changing in response to said subsequent connection between said one terminal and said KDC said priorly established terminal-unique cipher key.
- 15 2. The invention set forth in claim 1 wherein said session key is generated from the asymmetric exchange of information between said one terminal and said KDC plus the subsequent exchange of information between said first and second terminals.
- 20 3. The invention set forth in claim 2 wherein said session key at said one terminal is random with respect to information at said KDC.
4. The invention set forth in claim 2 wherein said session key at said one terminal is underivable with
- 25 respect to any information at said KDC.
5. A key distribution center for controlling the dissemination of session cipher keys between remotely located terminals, said center arranged for switched access to a plurality of said terminals, said center comprising
- 30 means for establishing communication cipher keys between said center and each said terminal having access thereto, each cipher key unique to each said terminal, means operative when one of said terminals accesses said center for bidirectional asymmetrically
- 35 exchanging information with said accessed terminal using, as a foundation for said exchange, said priorly established communication cipher keys, and

SUBSTITUTE SHEET



- 18 -

means responsive to said exchanged information for communicating to said terminal information allowing said terminal to establish a session cipher key for use with an identified other terminal also having access to said center.

6. The invention set forth in claim 5 wherein said key distribution center further comprising means for changing said established communication cipher keys as a result of said exchanged information.

7. The invention set forth in claim 5 wherein said cipher key establishing means uses information from a prior transmission from a particular terminal for establishing said cipher keys to said particular terminal.

8. The invention set forth in claim 5 wherein said exchanged information includes information generated in part at said center for the random generation of said session key allowing said session key to be underivable with respect to any information at said center.

9. A key distribution center for controlling the distribution of cipher control information among a number of terminals, said center comprising

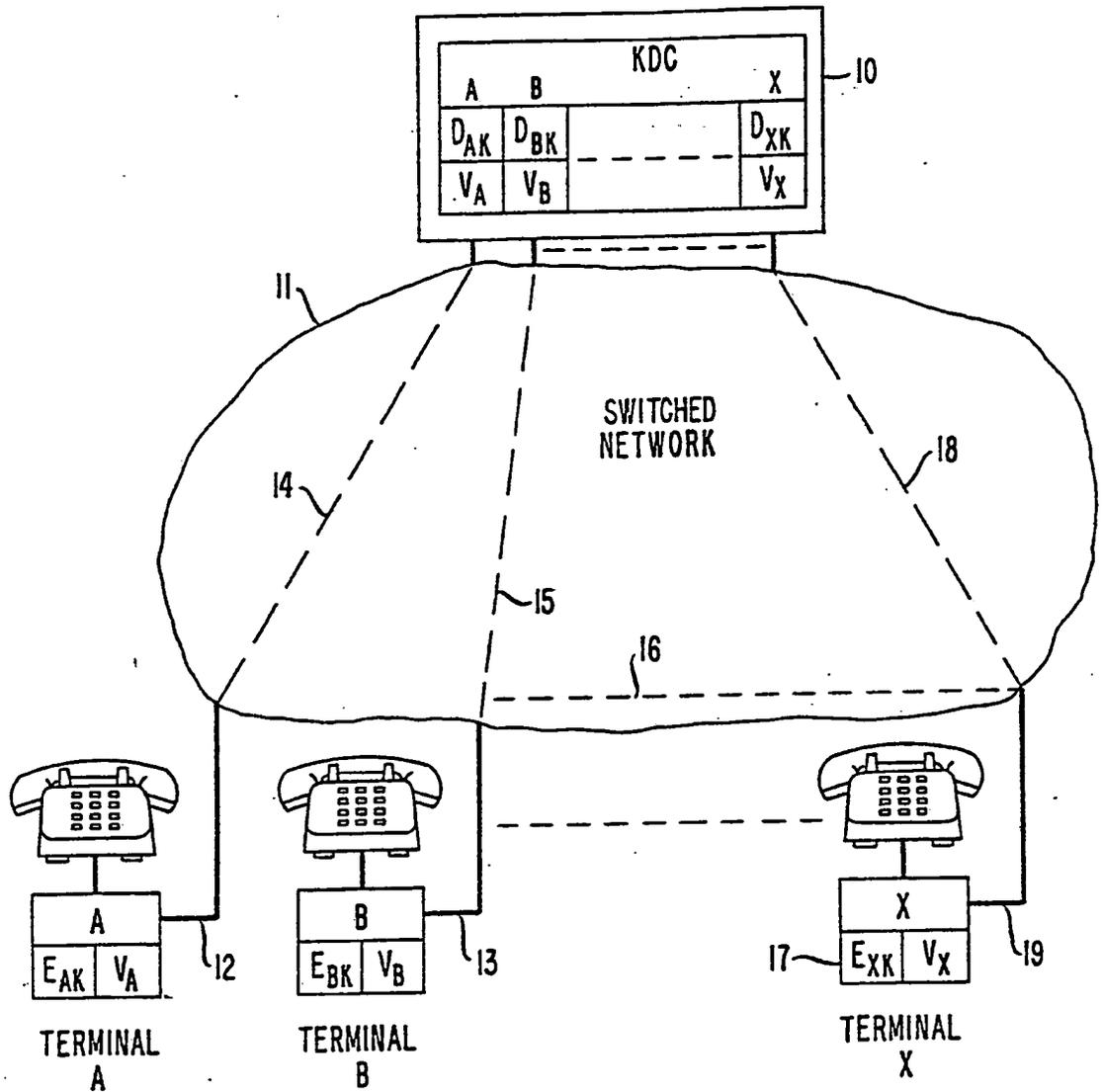
means for individually exchanging encoded information between any of said terminals, said exchange for any particular terminal based partially upon a last information exchange between said particular terminal and said center,

means for identifying at least two terminals where encrypted session information is to be exchanged and for accepting from said identified terminals certain encryption control information, and

means for modifying, according to a pre-established pattern, accepted information from said identified terminals and for communicating said modified information to the other of said terminals so as to allow each of said terminals to thereafter establish, independent of any information available at said center, a cipher key allowing said session information to be encrypted.

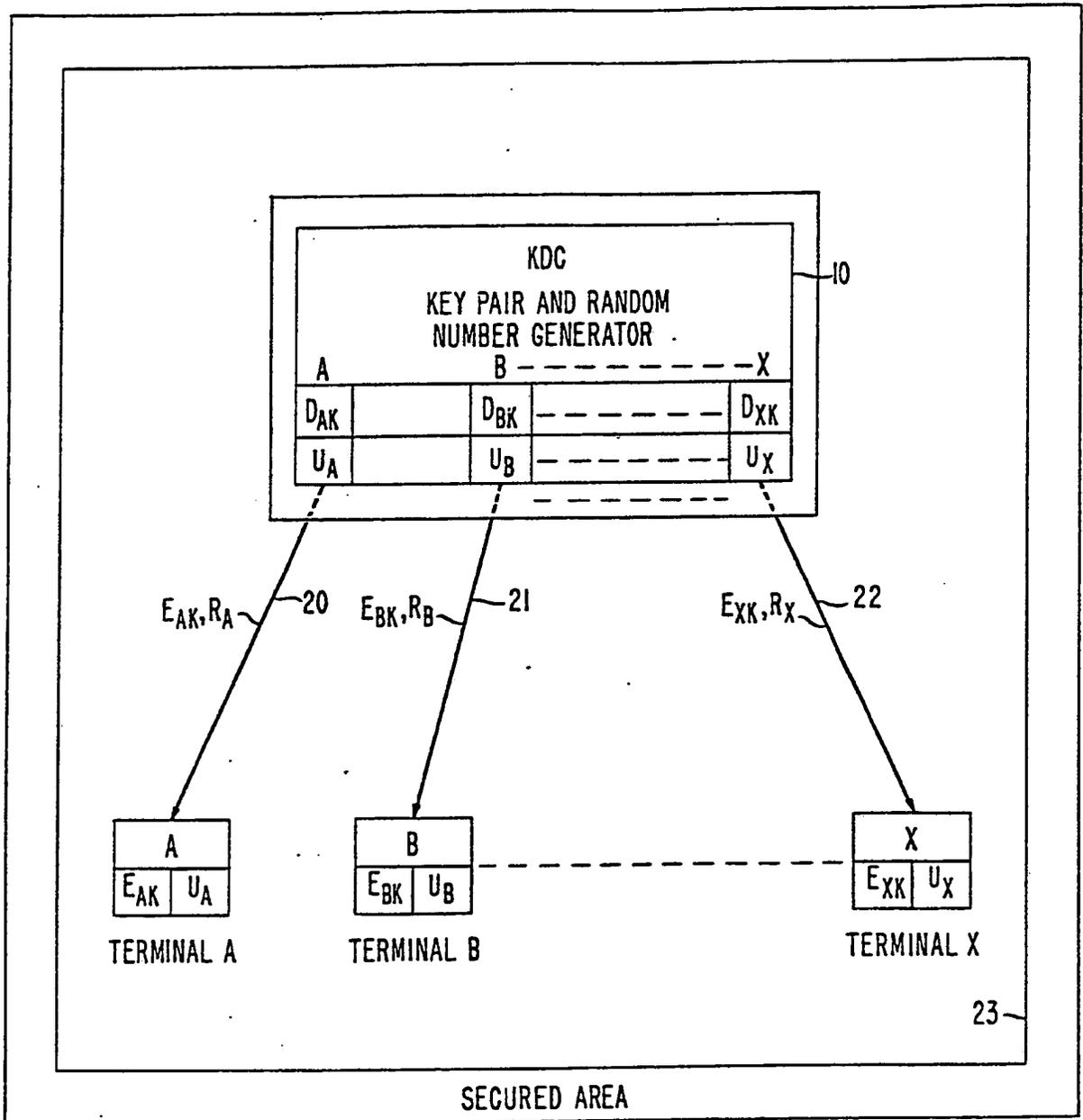
SUBSTITUTE SHEET

FIG. 1
KDC CONFIGURATION



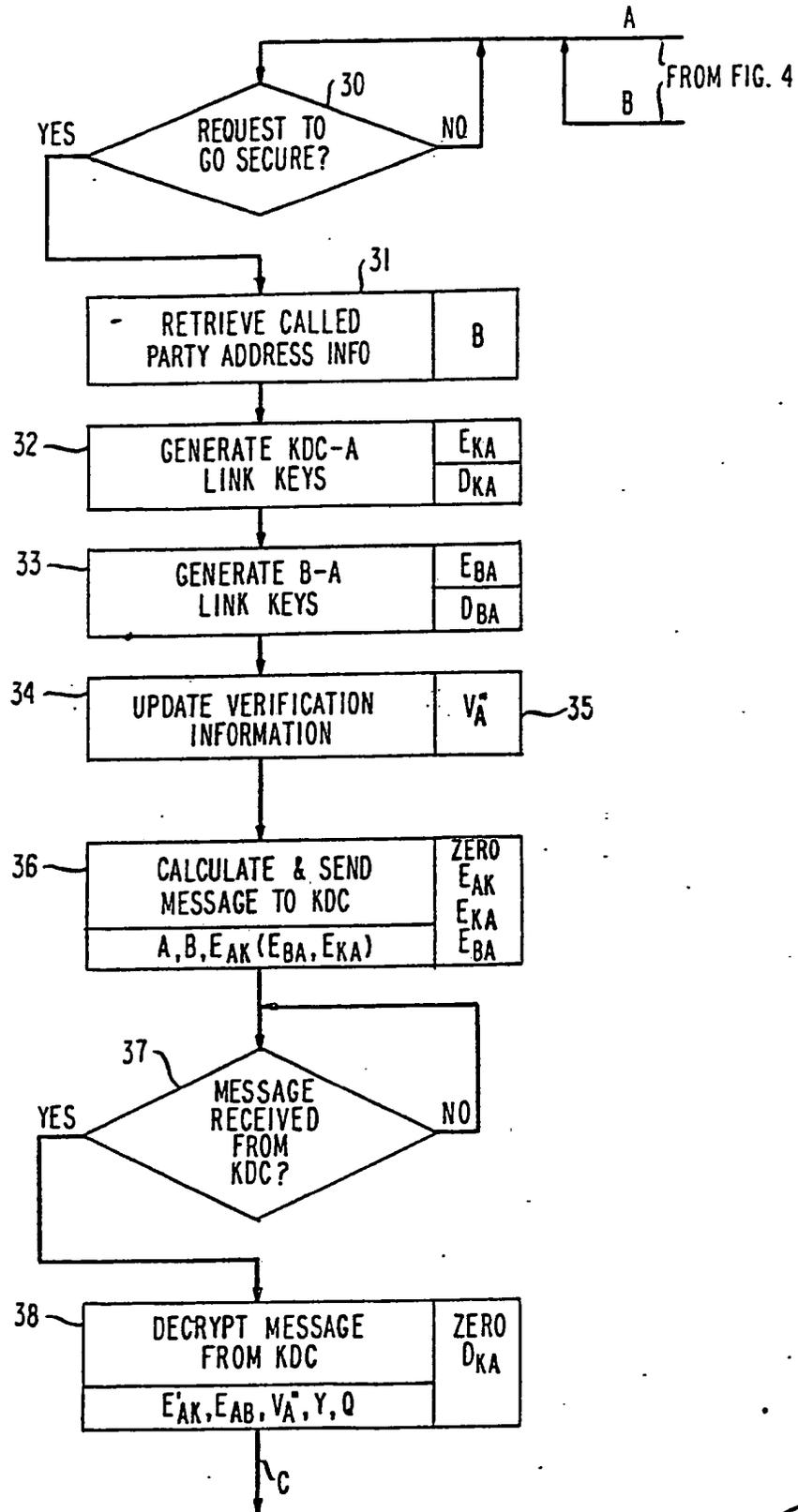
2/17

FIG. 2
INITIAL SYSTEM SETUP



3/17

FIG. 3
TERMINAL A



4/17

FIG. 4
TERMINAL A

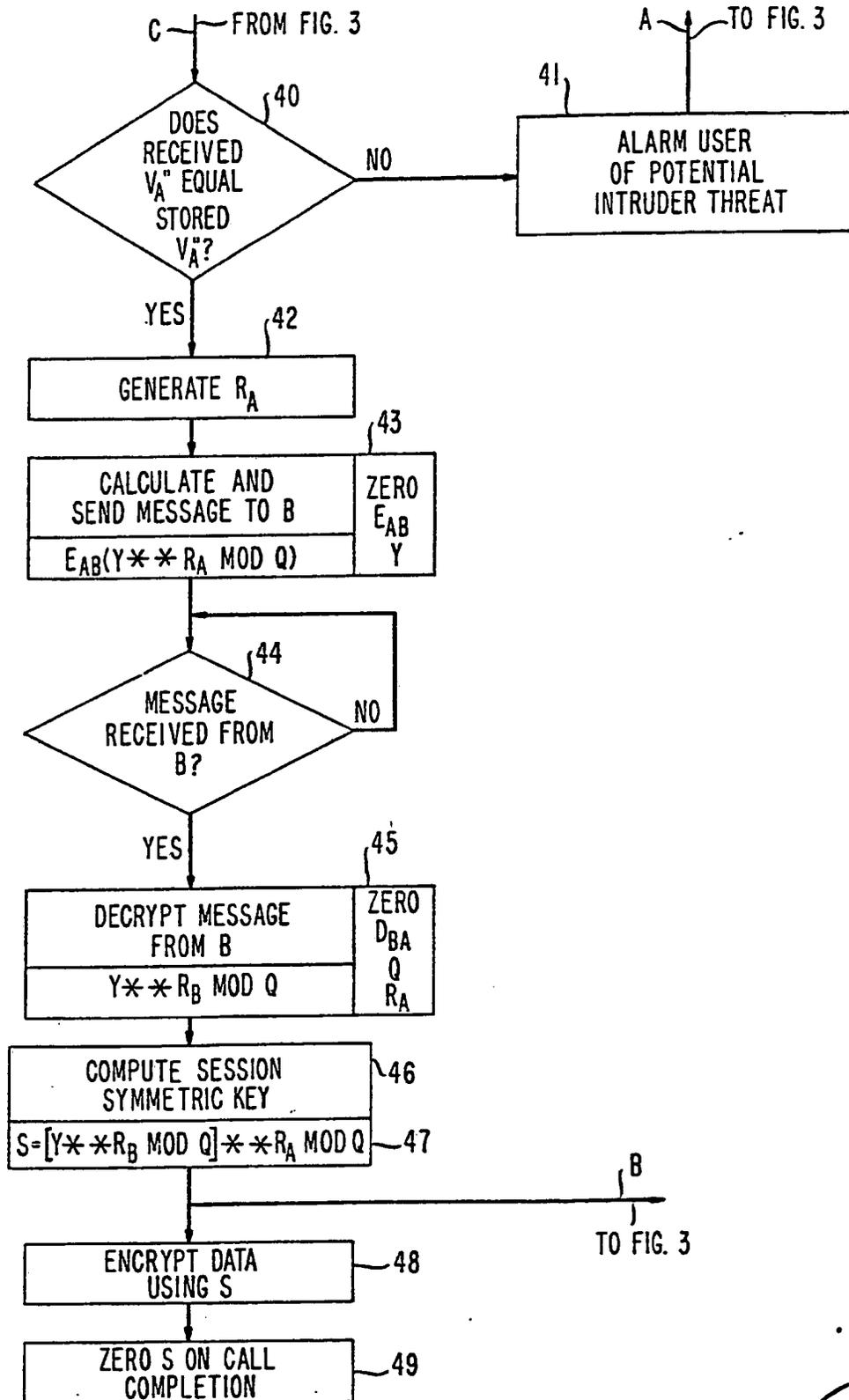
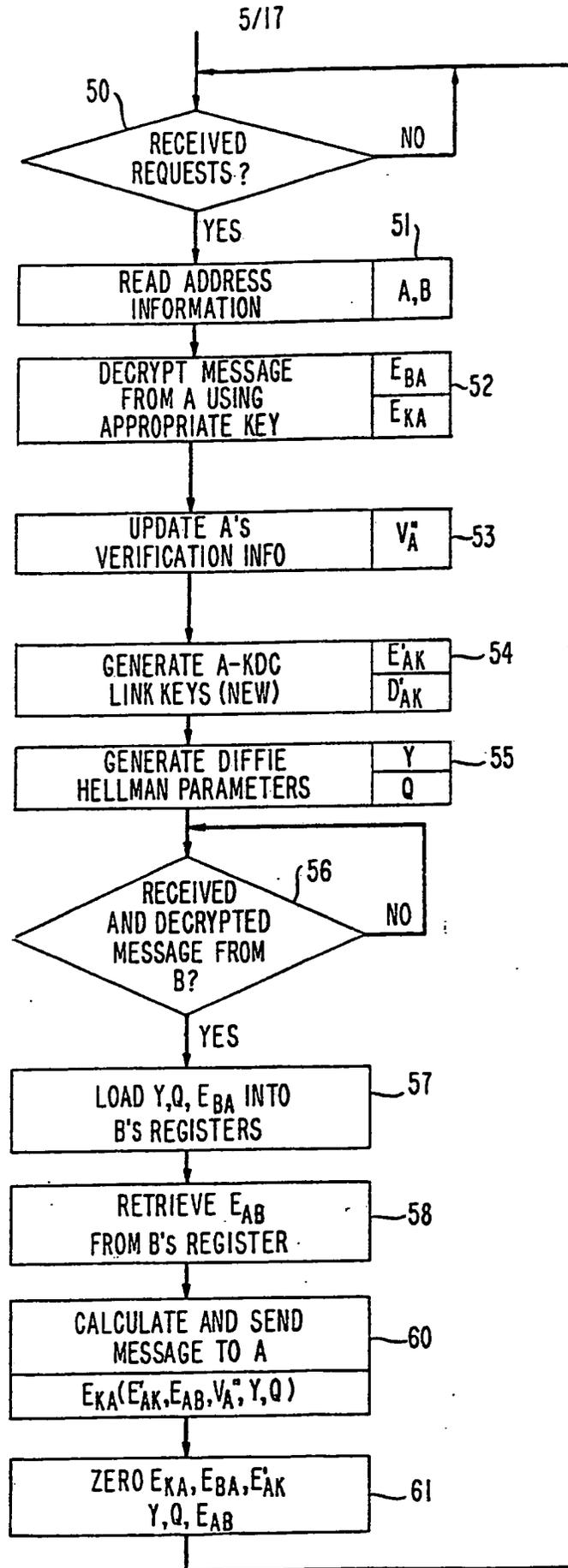


FIG. 5
KDC



6/17

FIG. 6 BETWEEN CALLS IDLE STATE (FOR TERMINAL A)

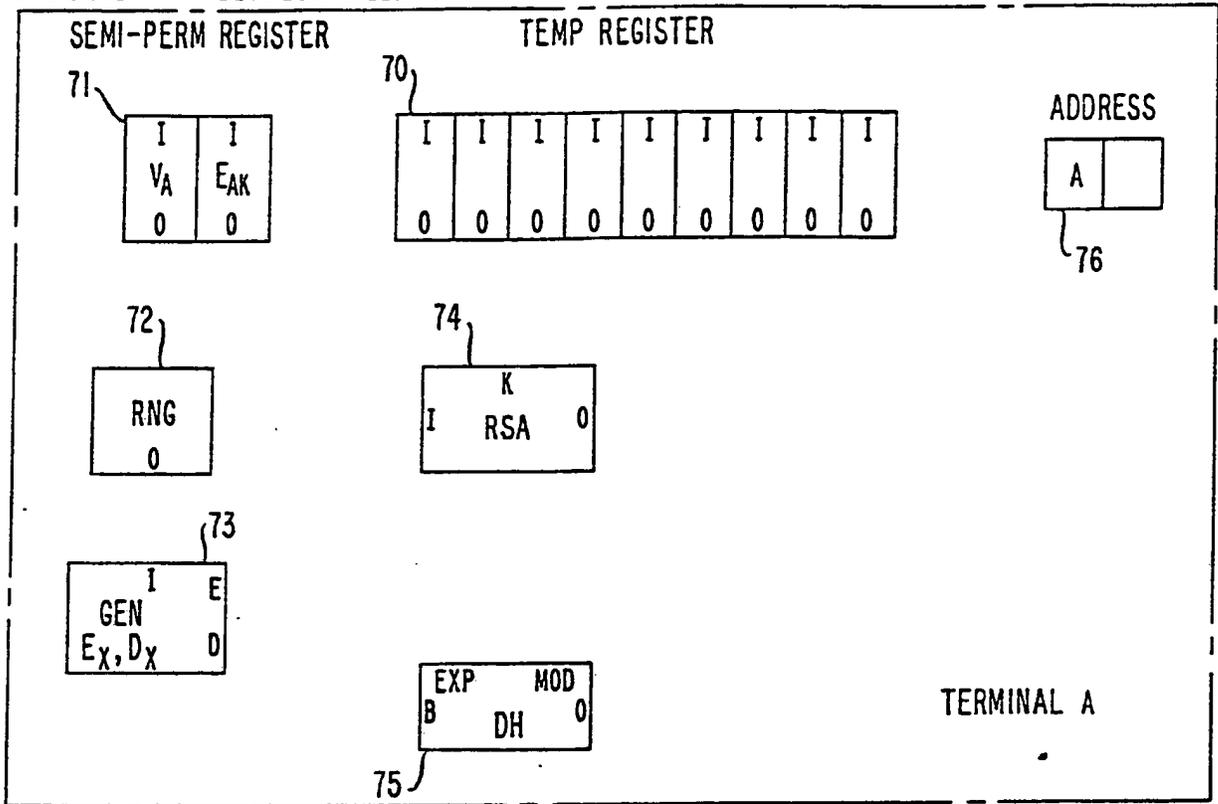
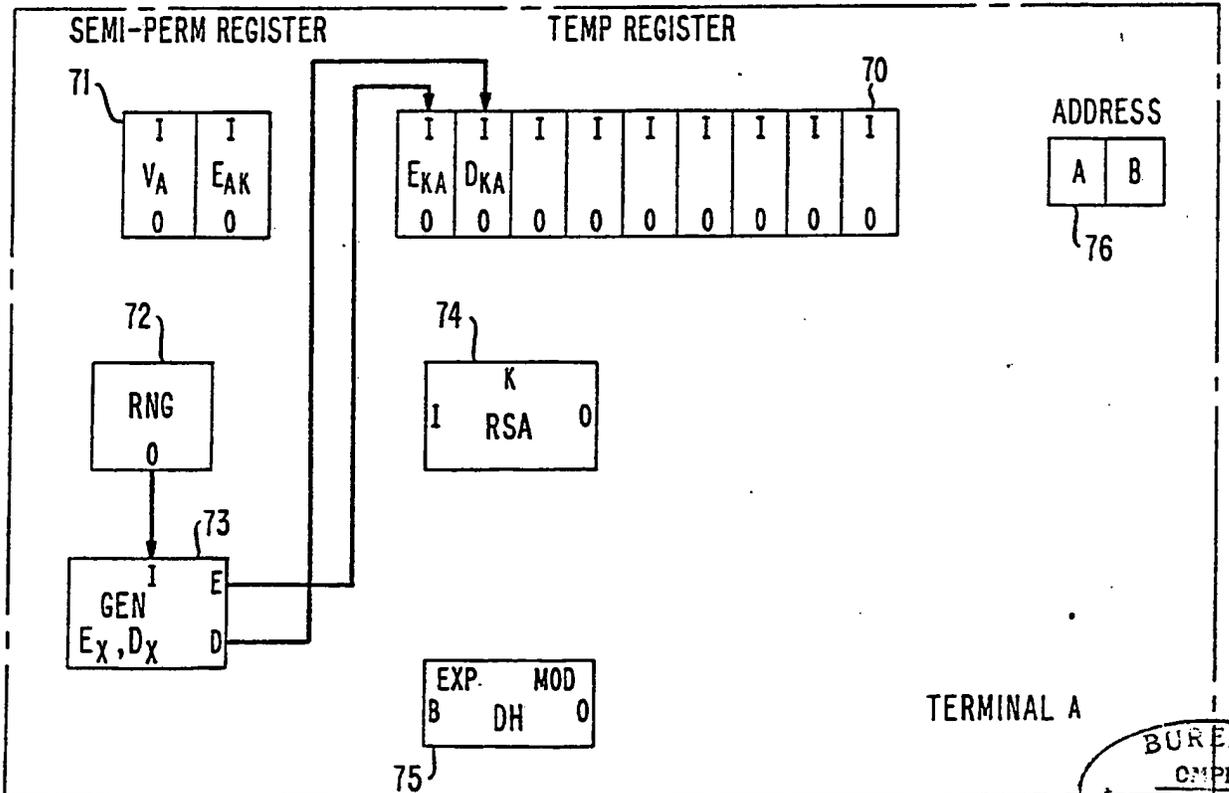


FIG. 7 START OF SECURE CALL (A TO B) SETUP - GENERATION OF KDC-A LINK KEYS



BUREAU
OMPI

FIG. 8 GENERATION OF B-A LINK KEYS

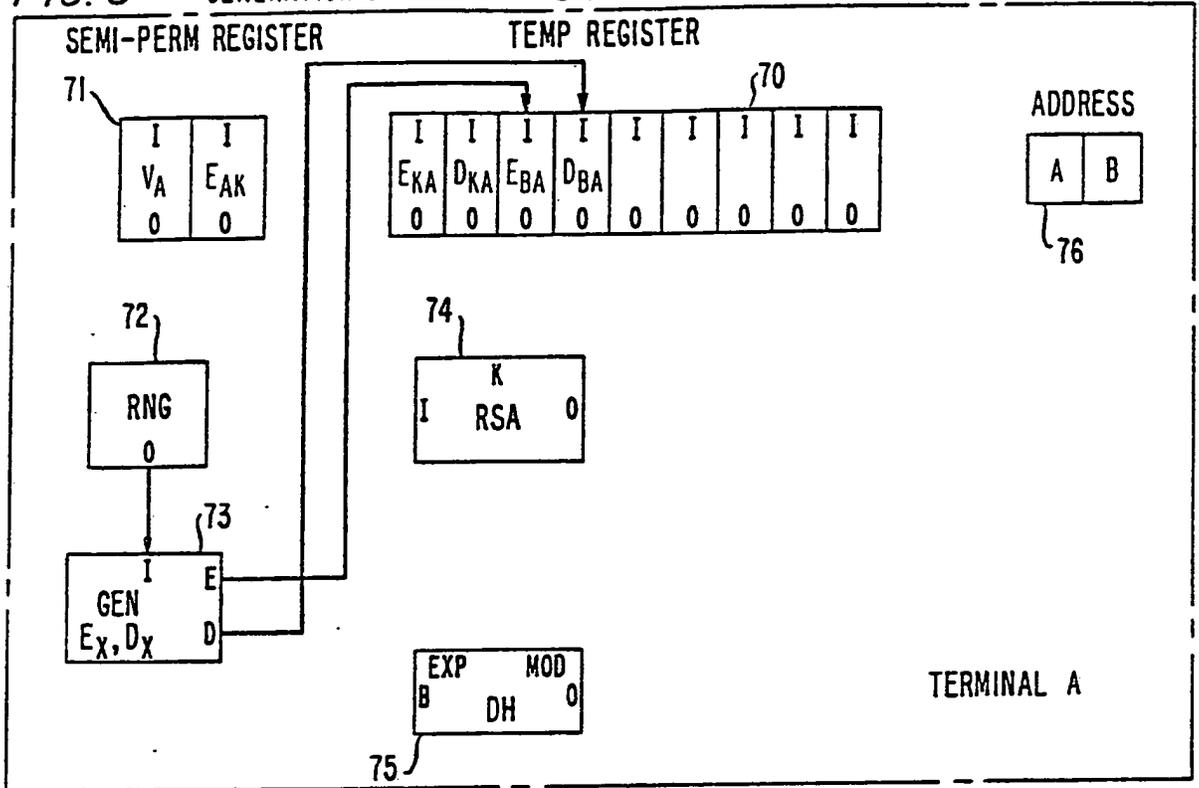
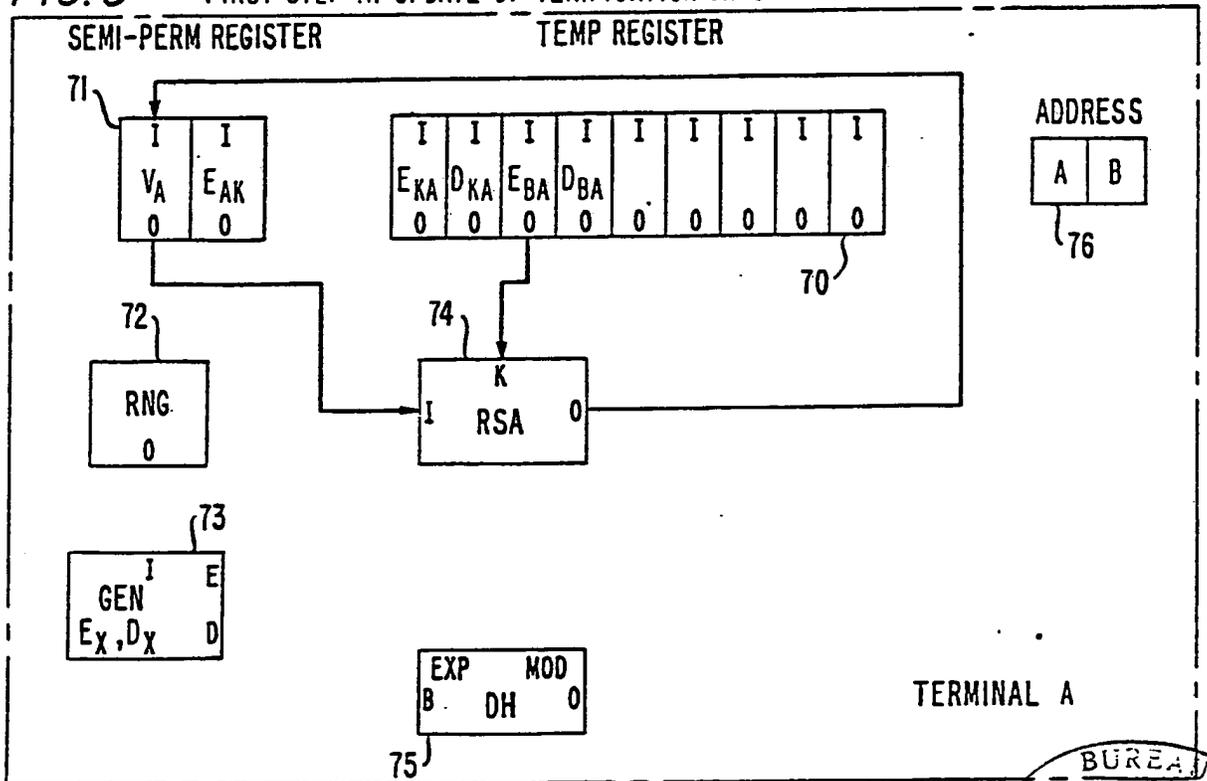


FIG. 9 FIRST STEP IN UPDATE OF VERIFICATION INFORMATION



BUREAU

FIG. 10 SECOND STEP IN UPDATE OF VERIFICATION INFORMATION

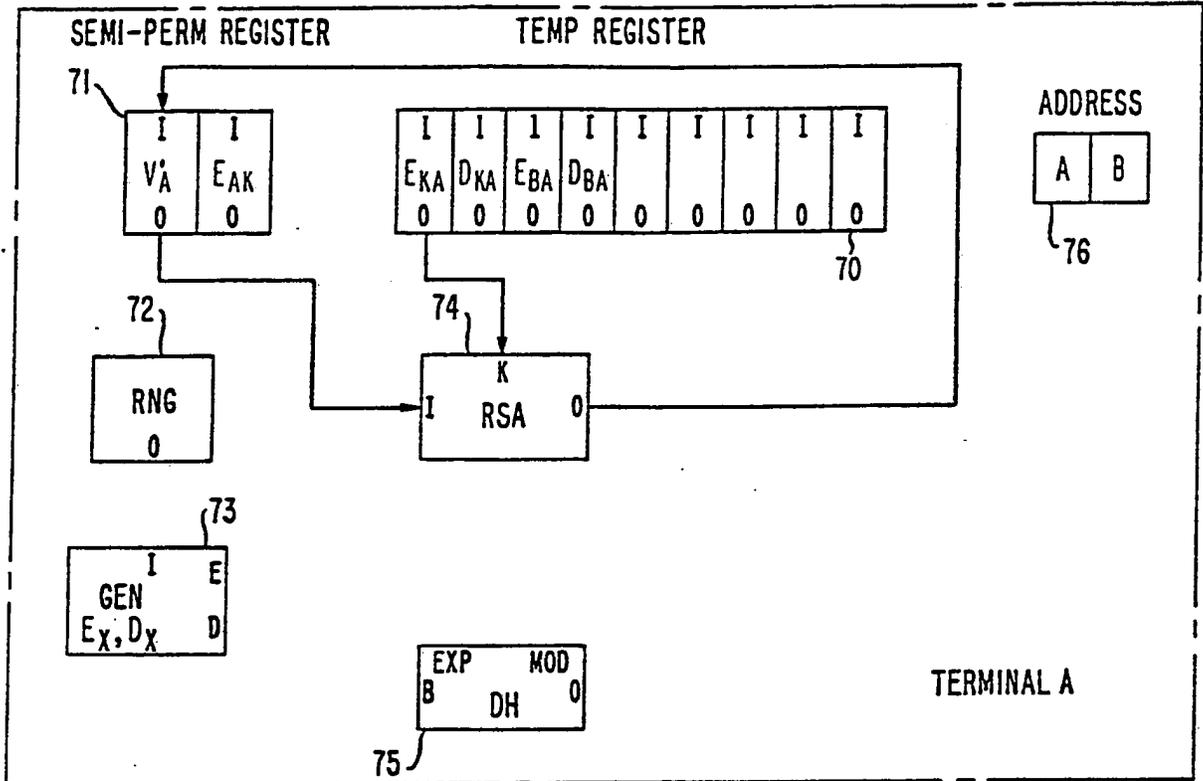
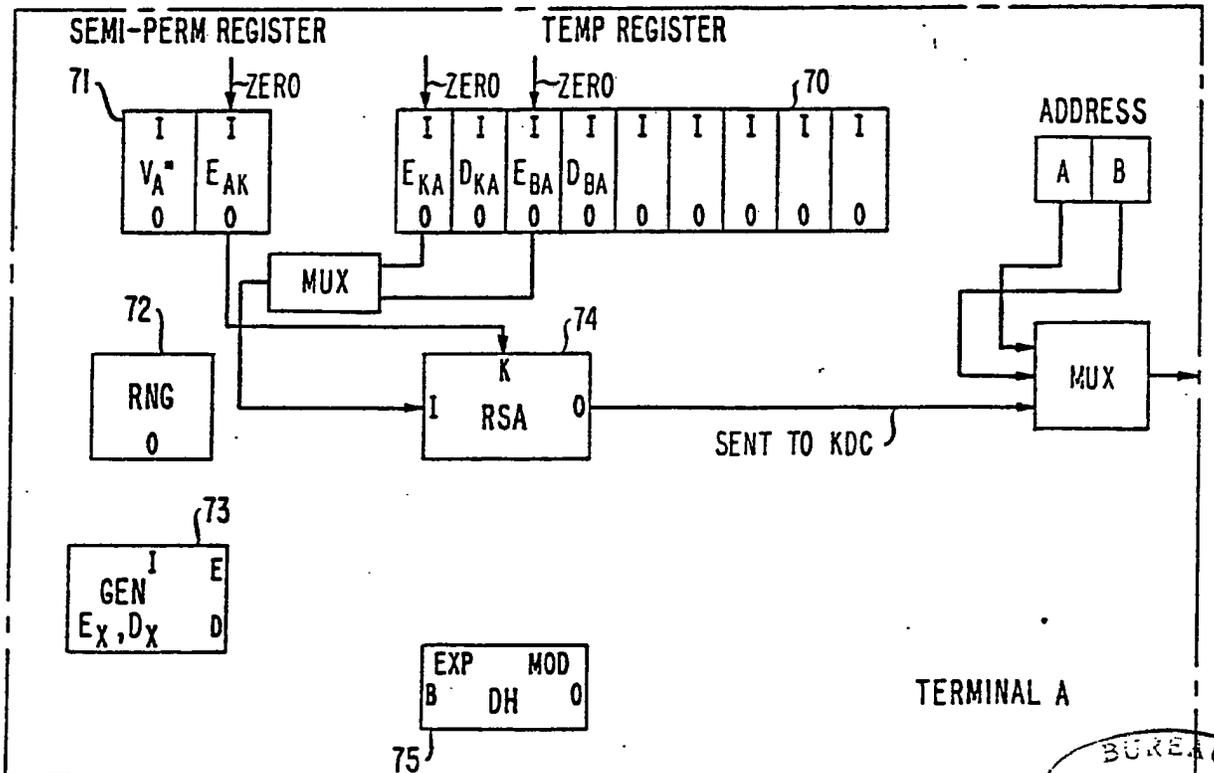


FIG. 11 COMPUTATION OF A-KDC MESSAGE



BUREAU

9/17

FIG. 12 IDLE STATE WHILE WAITING FOR RETURN MESSAGE FROM KDC

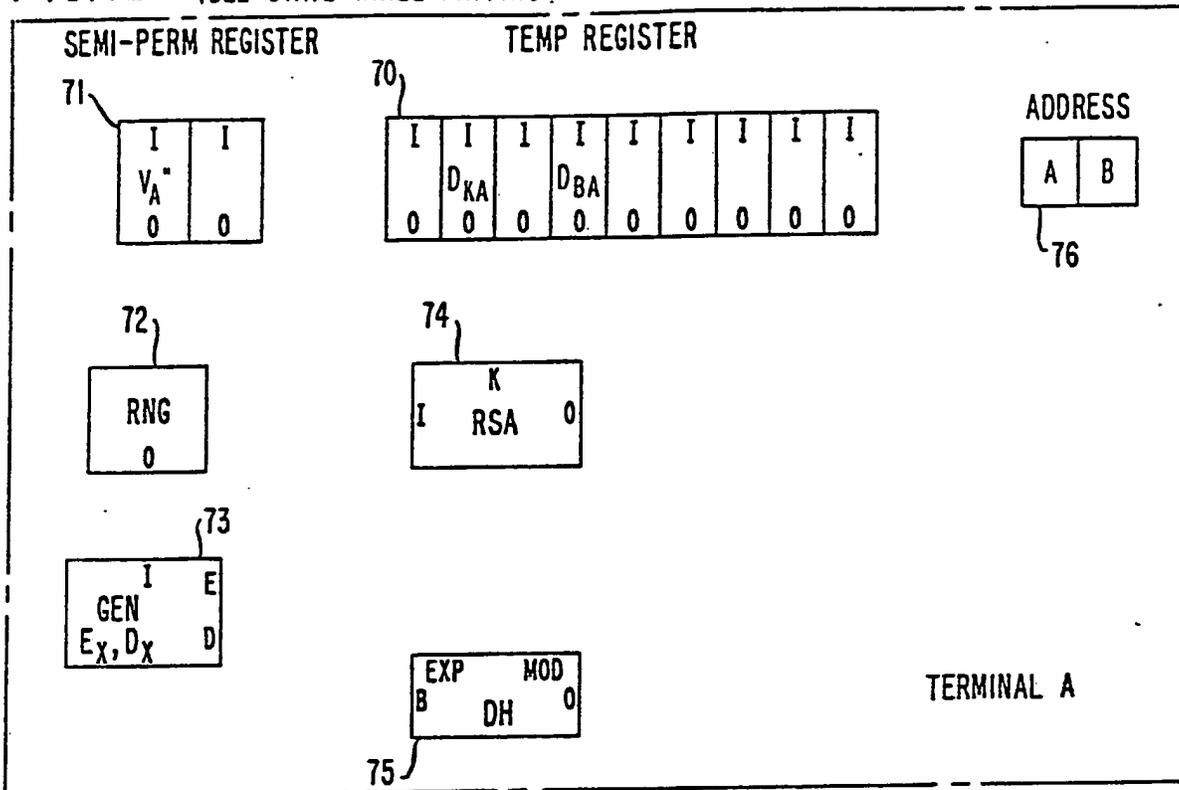
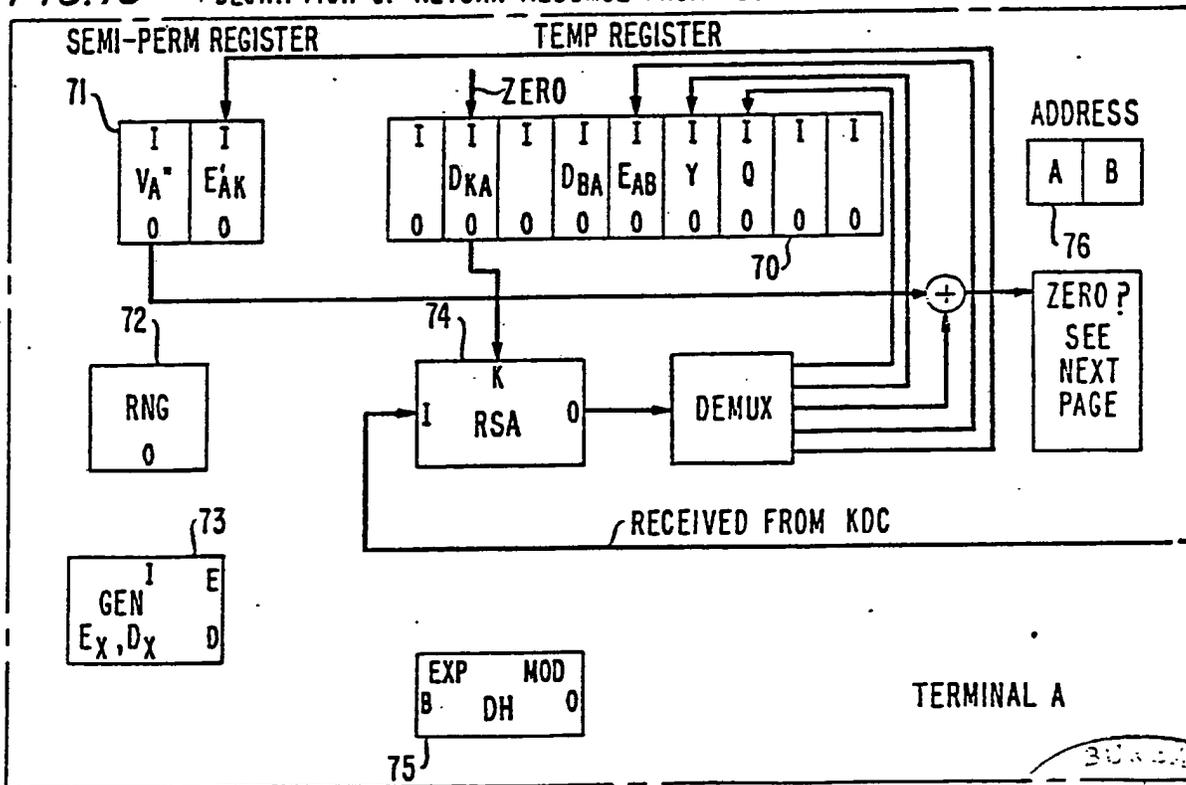


FIG. 13 DECRYPTION OF RETURN MESSAGE FROM KDC



10/17

FIG. 14 VERIFICATION CHECK

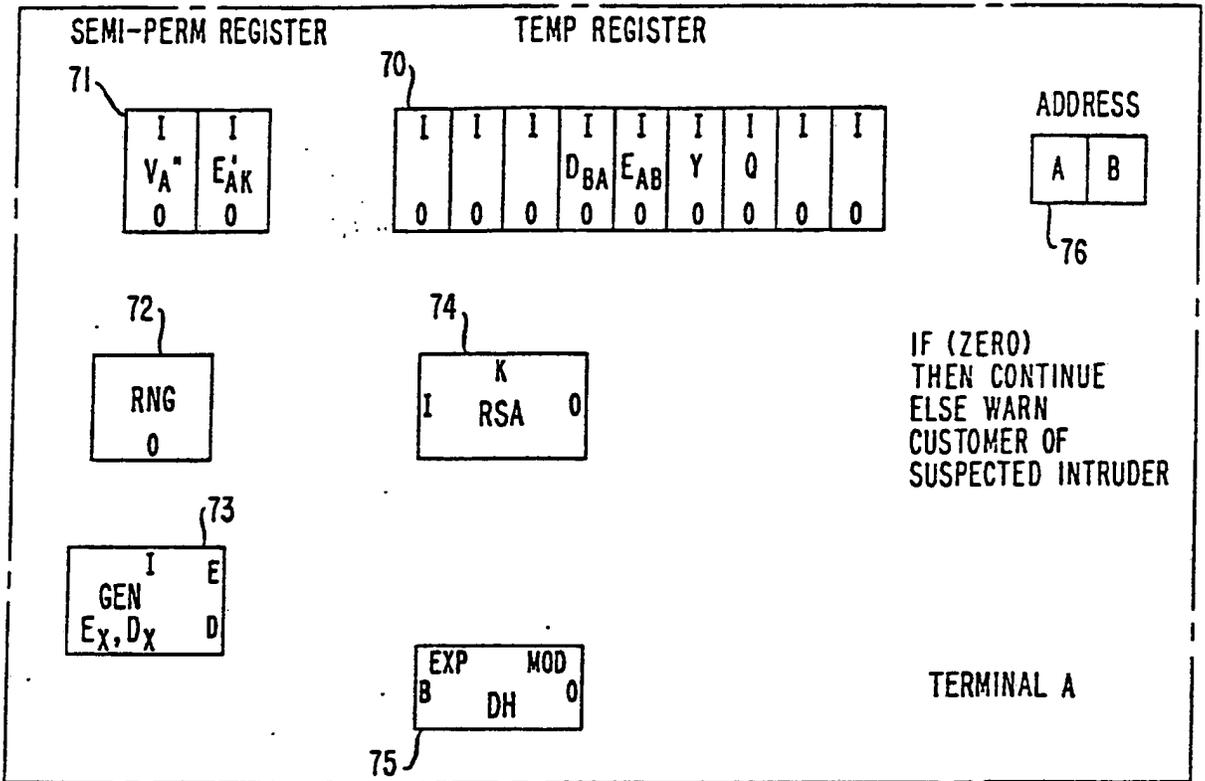
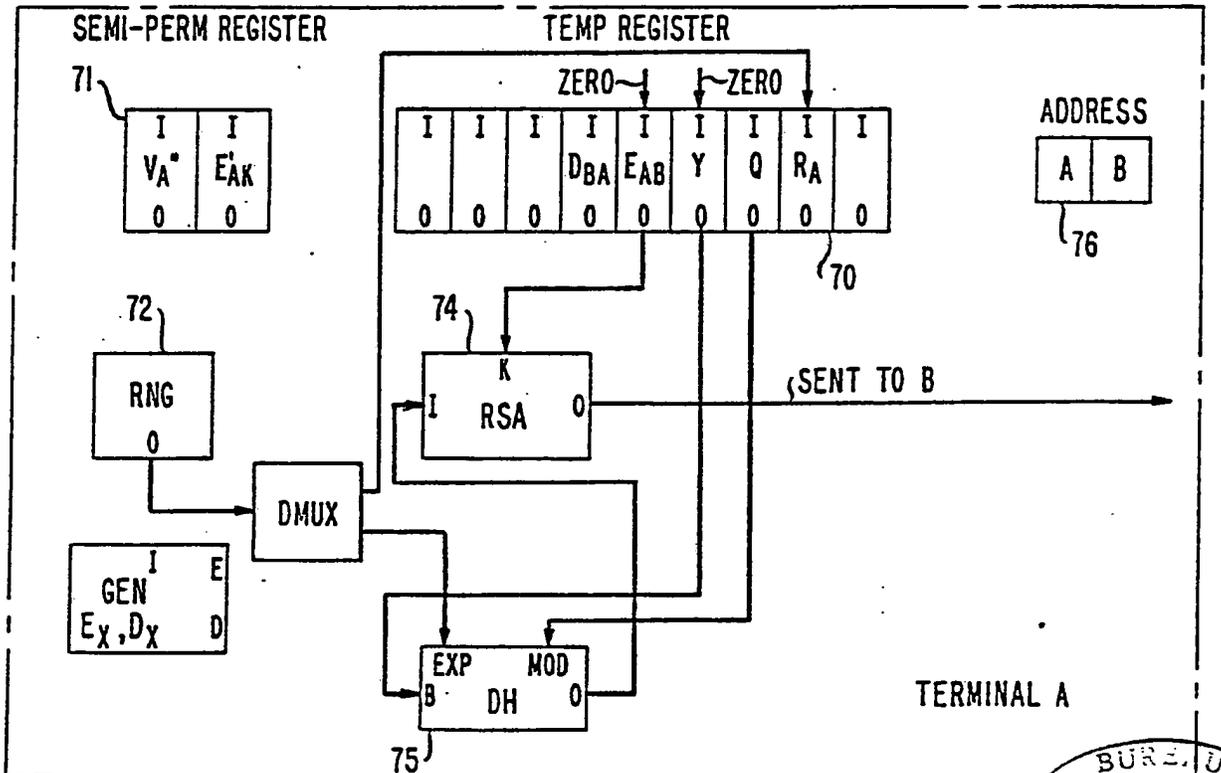


FIG. 15 START OF KEY EXCHANGE WITH B CALCULATION OF DIFFIE-HELLMAN KEYS



BUREAU

11/17

FIG. 16 IDLE WAIT STATE FOR RETURN MESSAGE FROM B

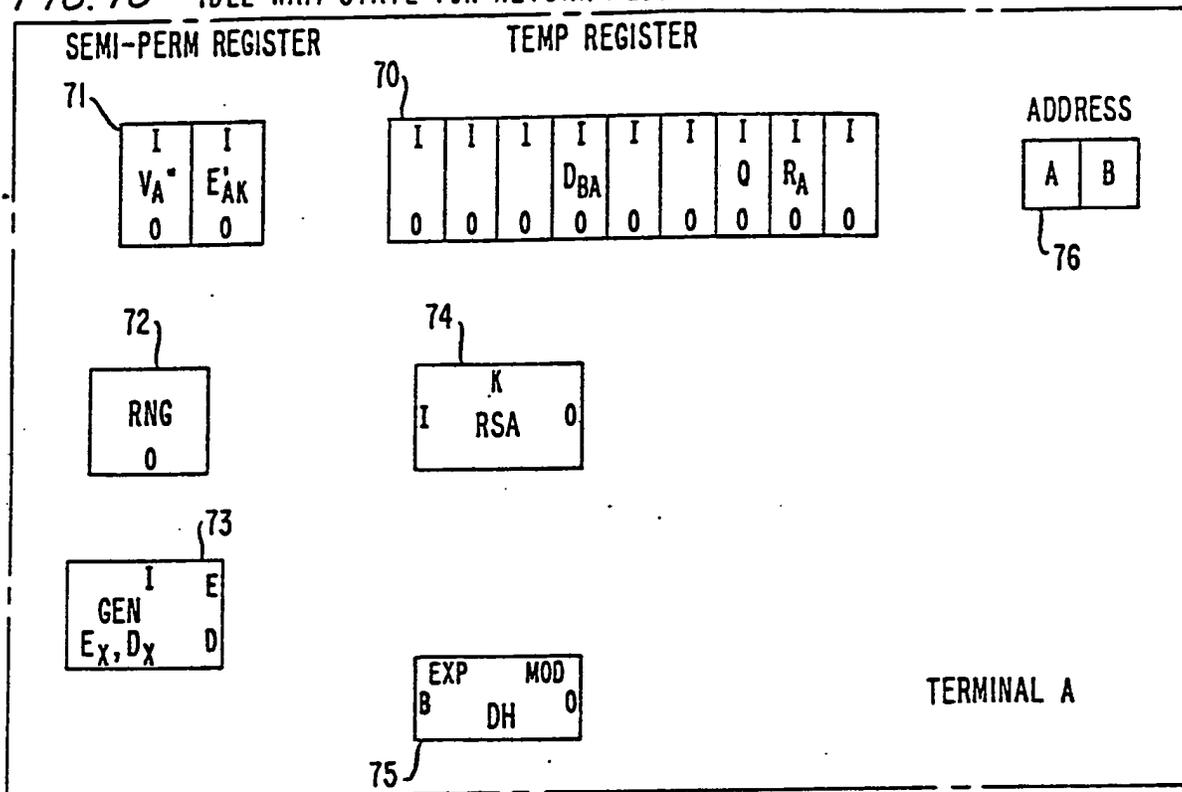
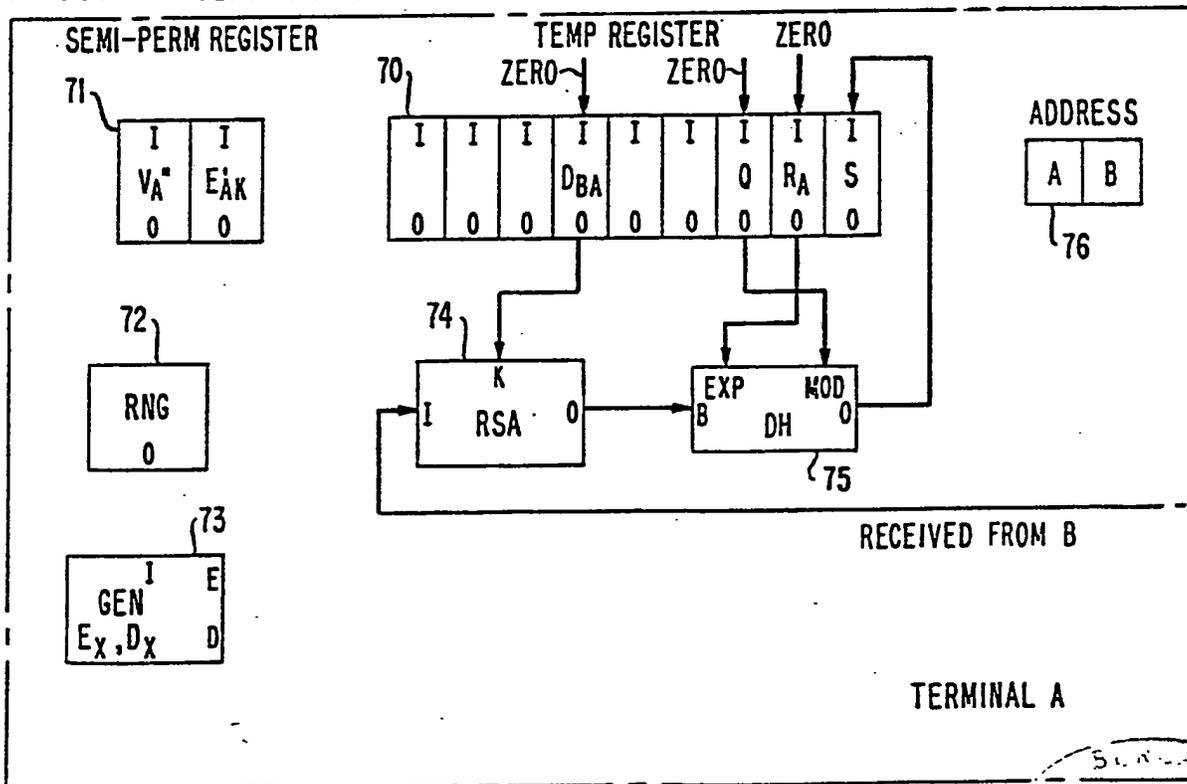


FIG. 17 DECRYPTION OF MESSAGE FROM B AND CALCULATION OF SESSION KEY-S



12/17

FIG. 18 KEY STORAGE DURING CALL

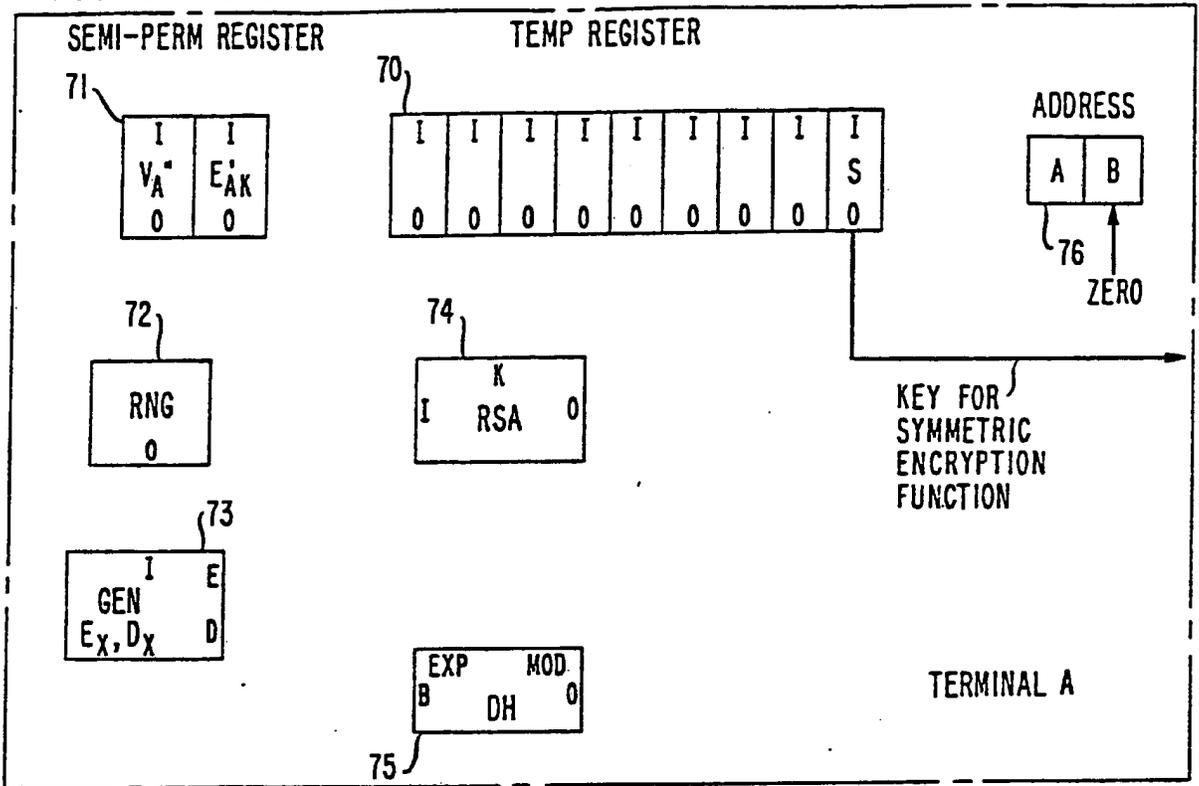


FIG. 19 IDLE STATE FOLLOWING CALL COMPLETION

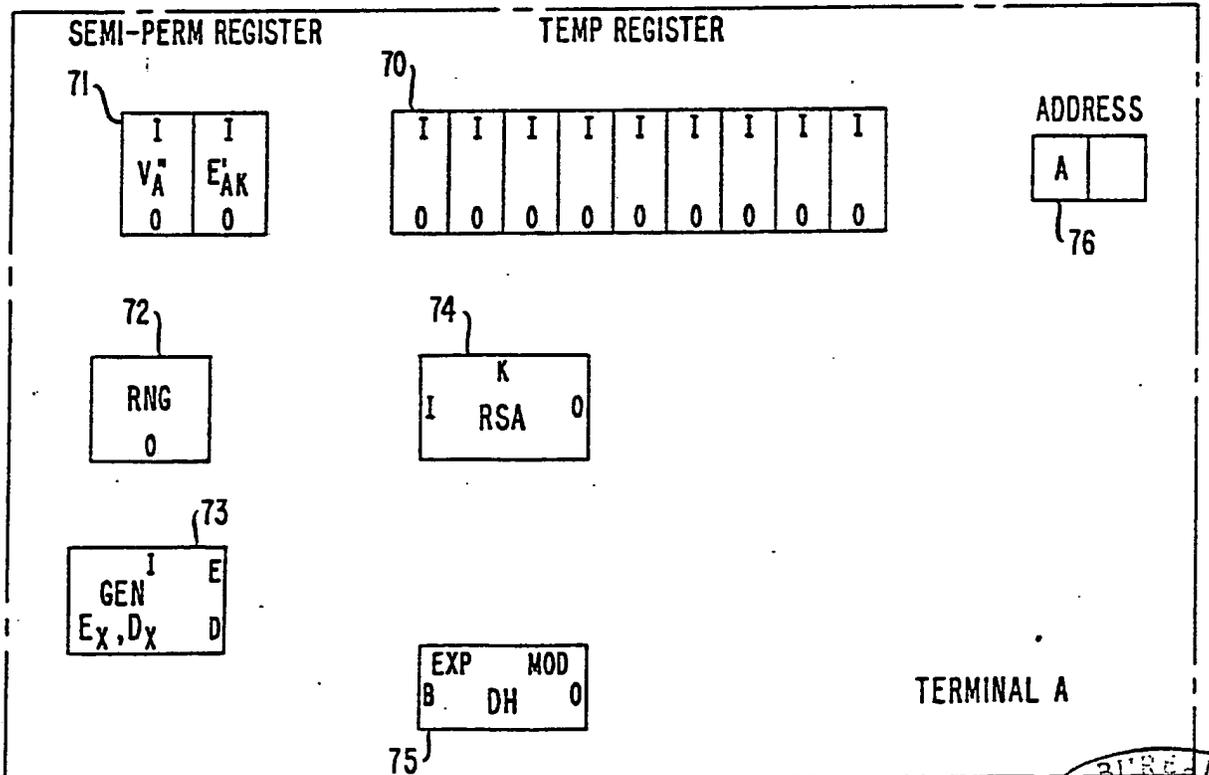


FIG. 20 IDLE STATE BETWEEN CALLS

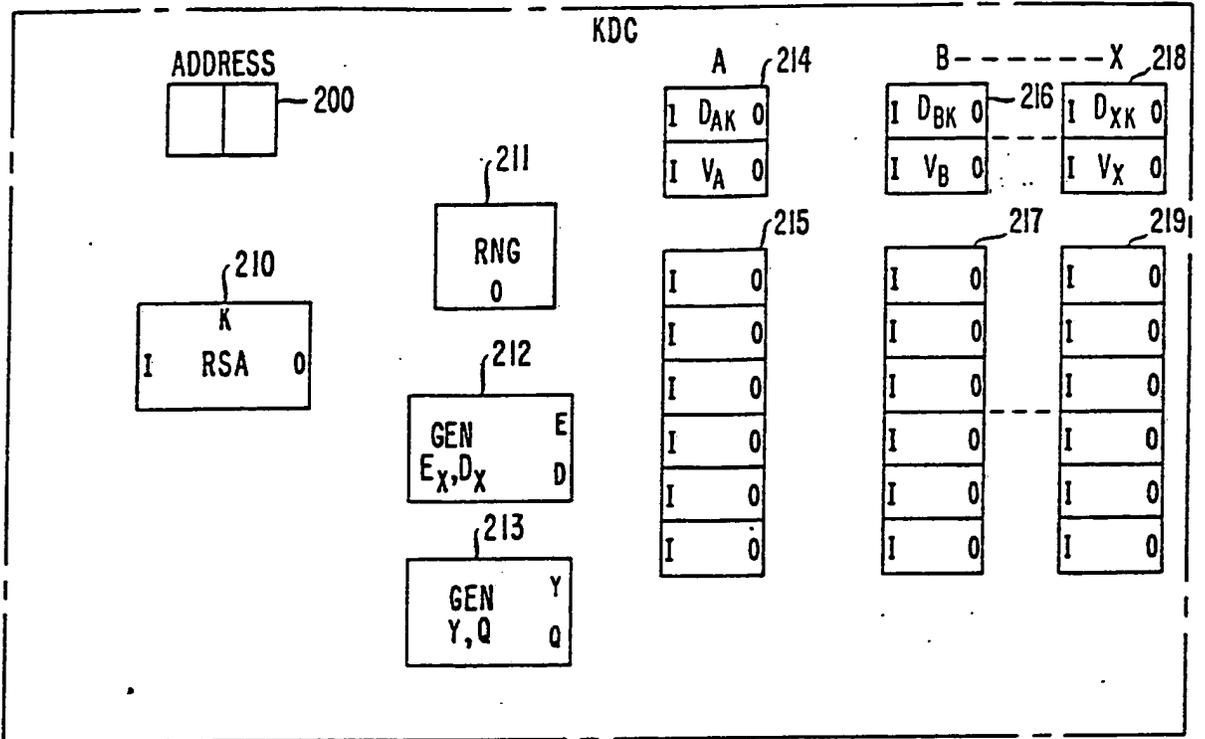


FIG. 21 DECRYPTION OF MESSAGE FROM A

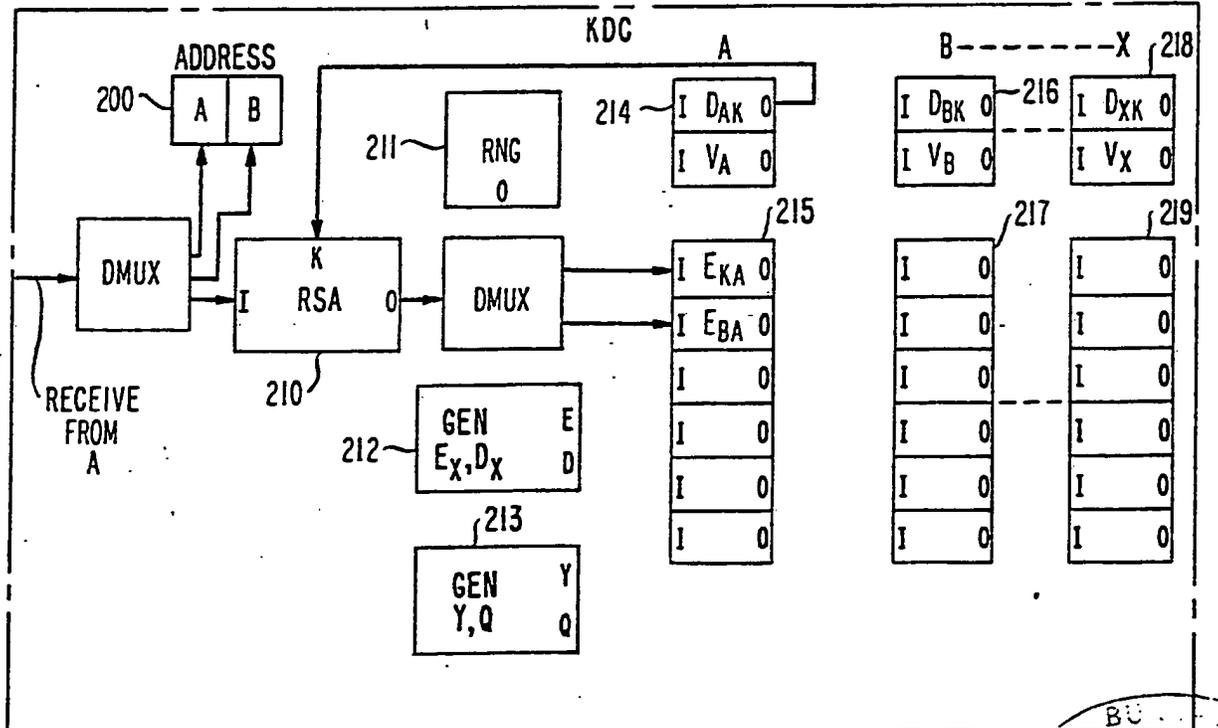


FIG. 22 FIRST STEP IN THE UPDATE OF VERIFICATION INFORMATION

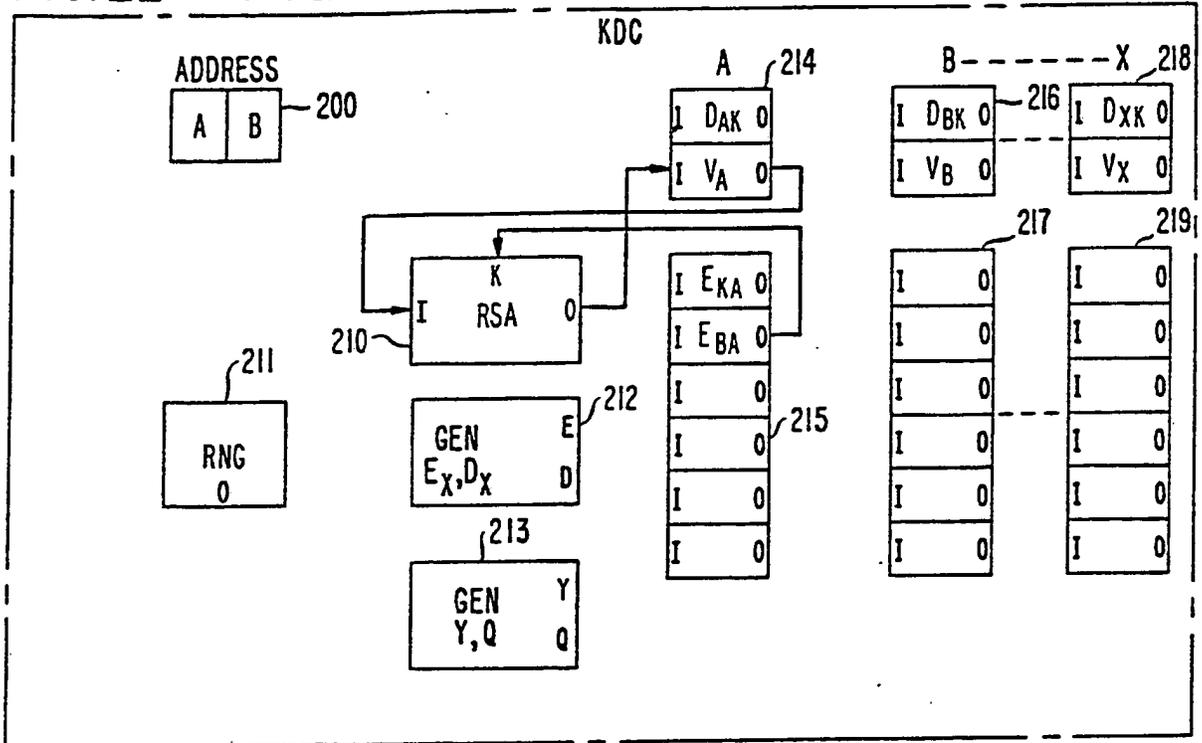
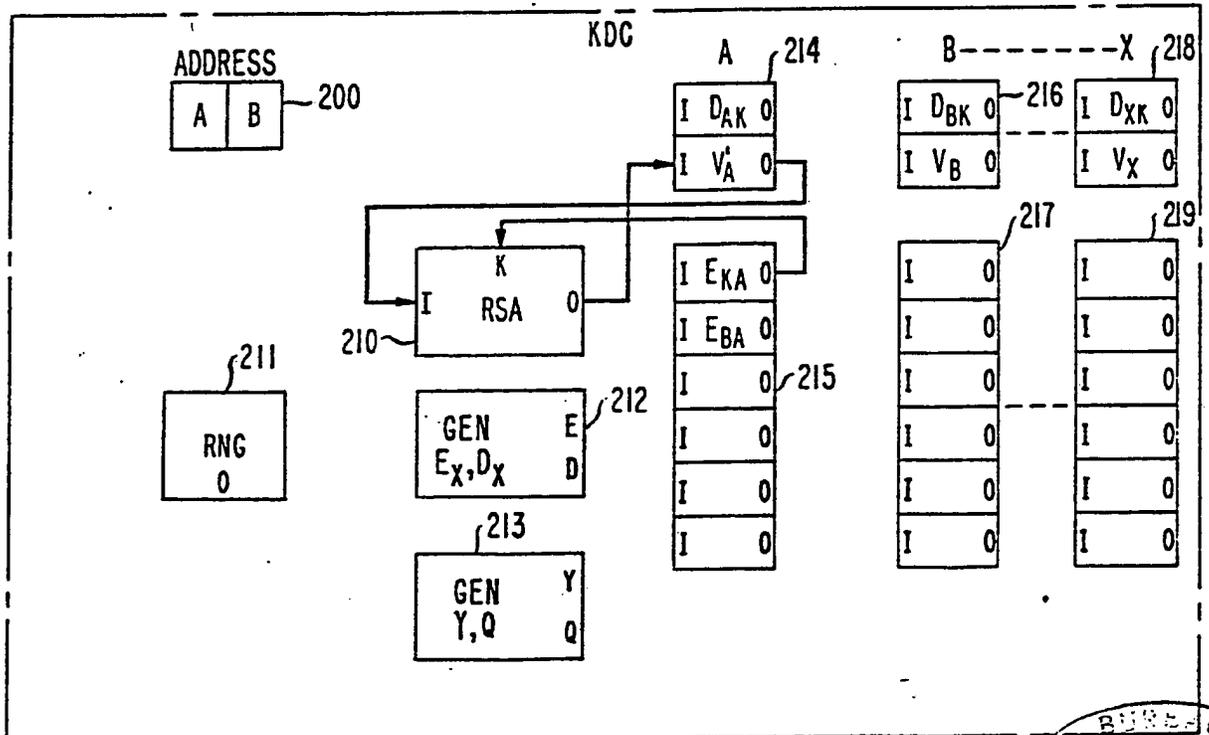


FIG. 23 SECOND STEP IN THE UPDATE OF VERIFICATION INFORMATION



BUREAU

FIG. 24 GENERATION OF NEW KDC-A LINK KEYS

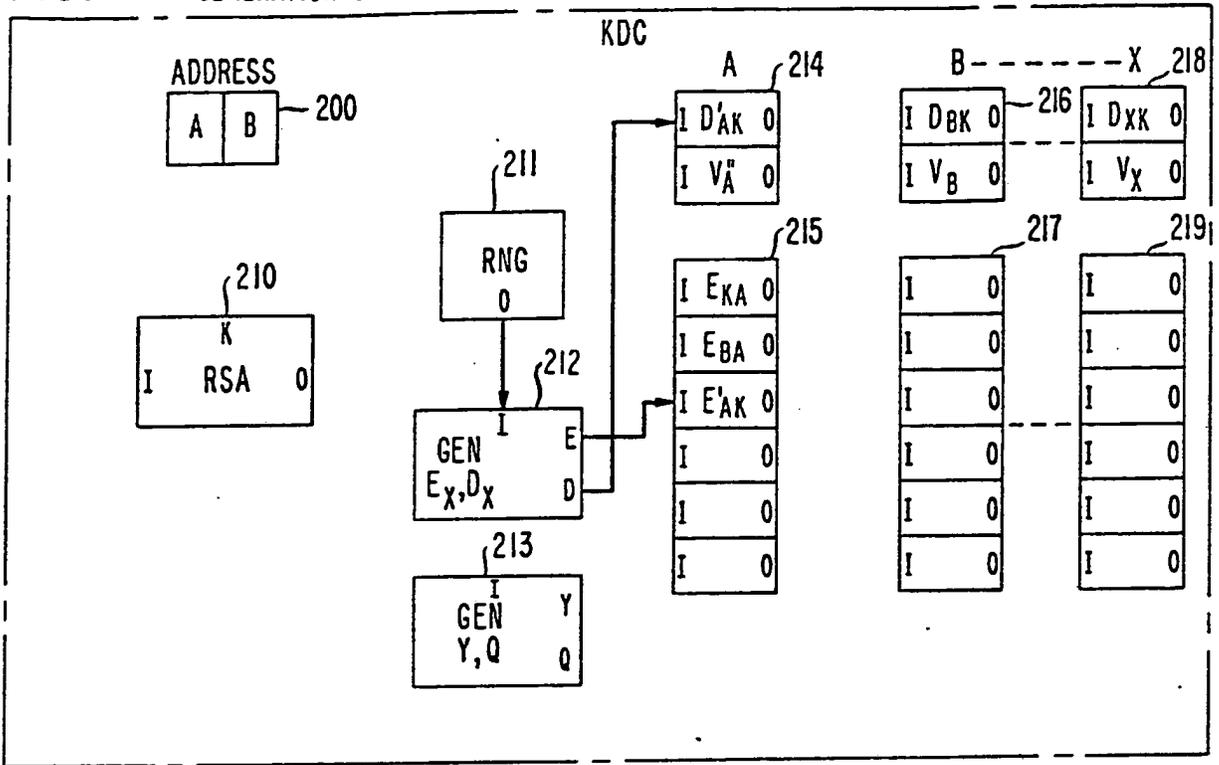


FIG. 25 GENERATION OF DIFFIE-HELLMAN ALGORITHM PARAMETERS

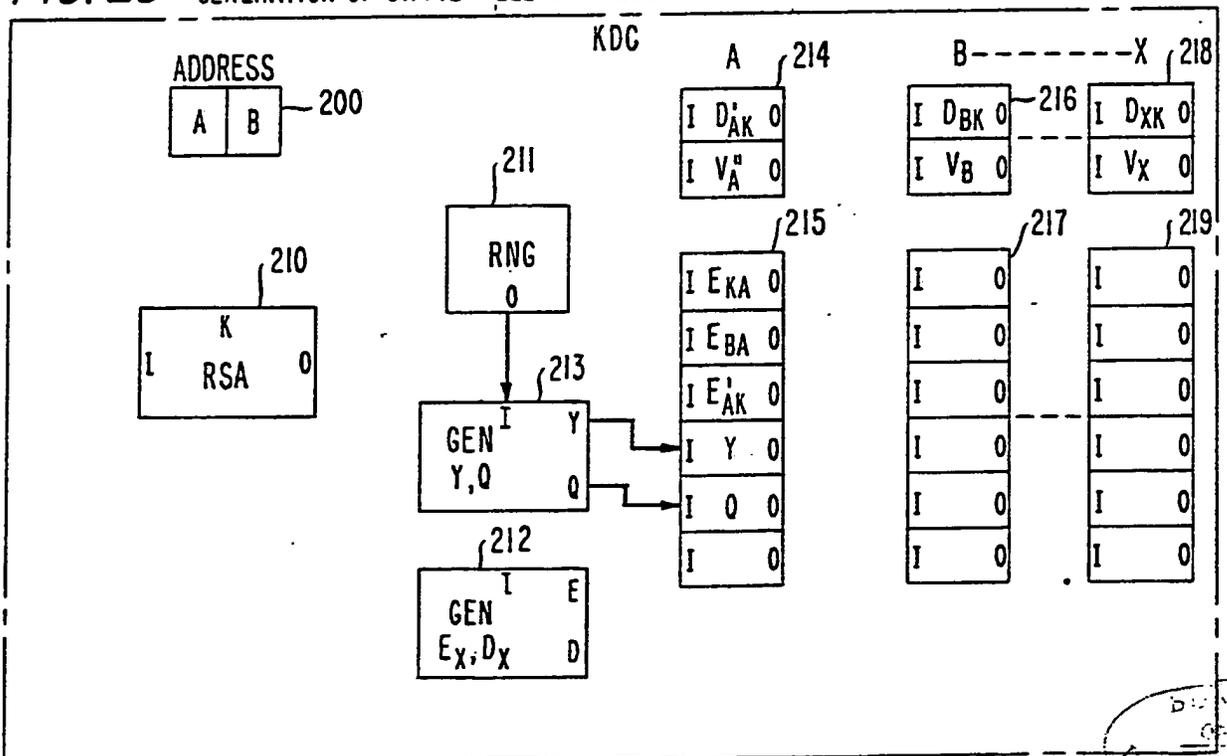


FIG. 26 INTERNAL EXCHANGE OF INFORMATION BETWEEN A & B'S REGISTERS

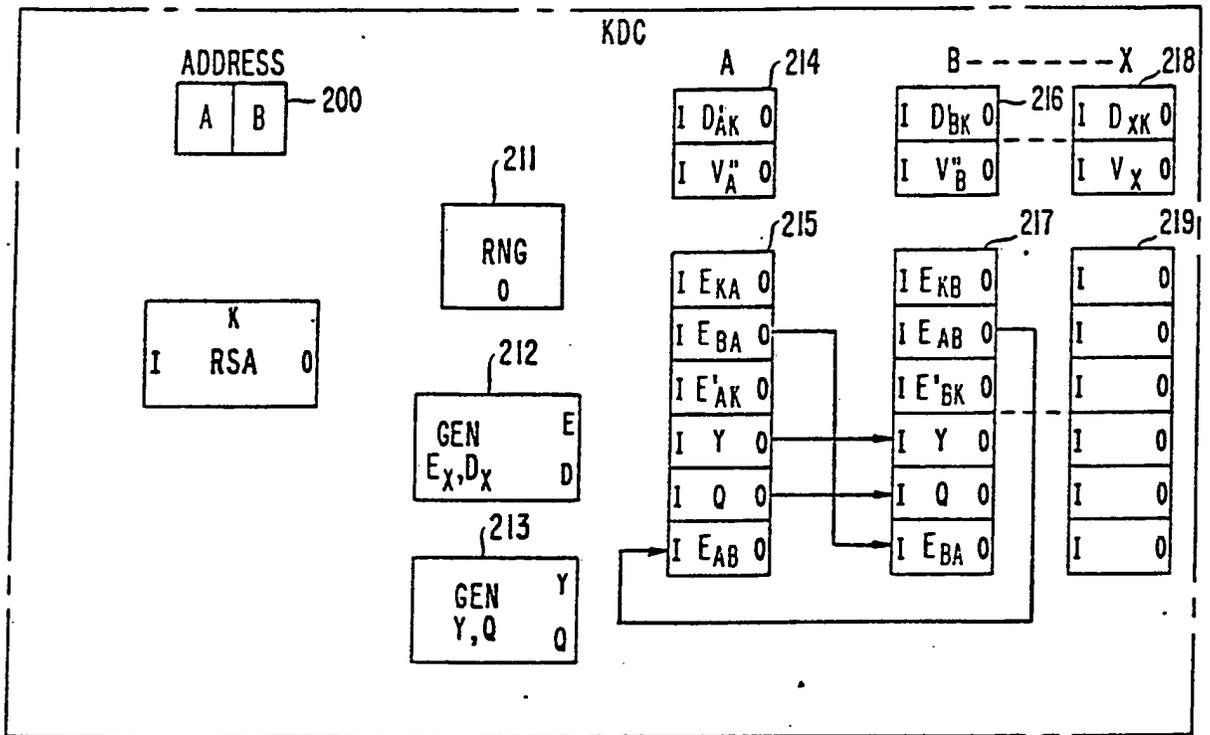


FIG. 27 COMPUTATION OF MESSAGE TO A

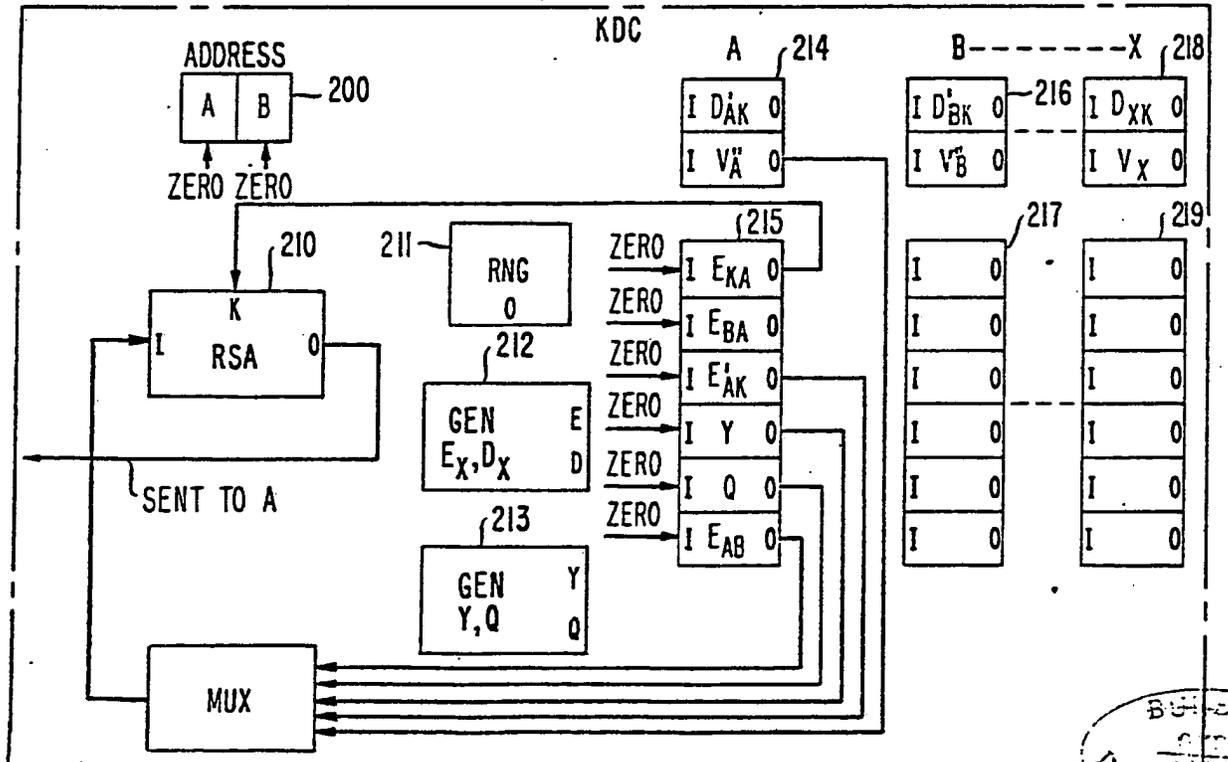
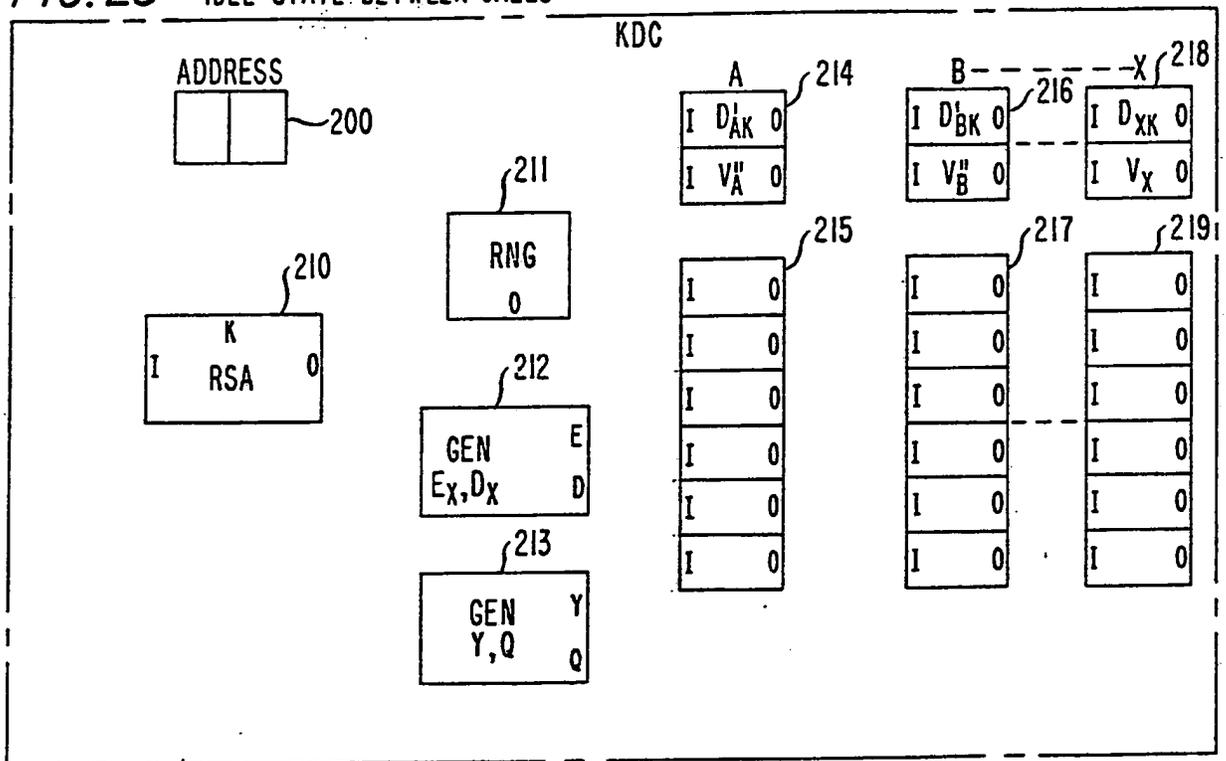
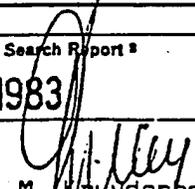


FIG. 28 IDLE STATE BETWEEN CALLS



INTERNATIONAL SEARCH REPORT

International Application No **PCT/US 83/00030**

I. CLASSIFICATION OF SUBJECT MATTER (If several classification symbols apply, indicate all) ³		
According to International Patent Classification (IPC) or to both National Classification and IPC		
IPC ³ : H 04 L 9/00		
II. FIELDS SEARCHED		
Minimum Documentation Searched ⁴		
Classification System	Classification Symbols	
IPC ³	H 04 L 9/00; H 04 L 9/02; H 04 K 1/00; H 04 L 9/04	
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched ⁵		
III. DOCUMENTS CONSIDERED TO BE RELEVANT ¹⁴		
Category ⁶	Citation of Document, ¹⁶ with indication, where appropriate, of the relevant passages ¹⁷	Relevant to Claim No. ¹⁸
A	EP, A1, 0048903 (LICENTIA) 7 April 1982 see page 2, line 23 - page 4, line 22 --	1, 2
A	US, A, 4182933 (ROSENBLUM) 8 January 1980 see column 7, lines 1-21; column 8, lines 37-50; column 11, lines 17-41 cited in the application --	1
A	DATAMATION, vol. 22, no. 8, August 1976 (Barrington, US) Sykes: "Protecting data by encryption", pages 81-85, see page 84, left-hand column, lines 23-42 --	1
A	IBM-Technical Disclosure Bulletin, vol. 22, no. 2, July 1979 (New York, US) Lennon et al.: "Composite cryptographic session keys for enhanced communication security", pages 643-646, see page 643, first line to page 644, line 8 --	1, 2
A	Fifth International Conference on Digital Satellite Communications, 23-26 March 1981 (New York, US) Bic et al.:	./.
<p>⁶ Special categories of cited documents: ¹⁵</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"A" document member of the same patent family</p>		
IV. CERTIFICATION		
Date of the Actual Completion of the International Search ³	Date of Mailing of this International Search Report ³	
20th April 1983	03 MAI 1983	
International Searching Authority ¹	Signature of Authorized Officer ²⁰	
EUROPEAN PATENT OFFICE	 G.L.M. Huydenberg	

FURTHER INFORMATION CONTINUED FROM THE SECOND SHEET	
<p>"Privacy over digital satellite links", pages 243-249, see page 246, right-hand column, lines 32-43; page 247, left-hand column, lines 6-11; figure 3</p> <p style="text-align: center;">-----</p>	<p>1</p>

V. OBSERVATIONS WHERE CERTAIN CLAIMS WERE FOUND UNSEARCHABLE ¹⁰

This international search report has not been established in respect of certain claims under Article 17(2) (a) for the following reasons:

1. Claim numbers _____, because they relate to subject matter ¹² not required to be searched by this Authority, namely:

2. Claim numbers _____, because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out ¹³, specifically:

VI. OBSERVATIONS WHERE UNITY OF INVENTION IS LACKING ¹¹

This International Searching Authority found multiple inventions in this international application as follows:

1. As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims of the international application.

2. As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims of the international application for which fees were paid, specifically claims:

3. No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claim numbers:

4. As all searchable claims could be searched without effort justifying an additional fee, the International Searching Authority did not invite payment of any additional fee.

Remark on Protest

The additional search fees were accompanied by applicant's protest.

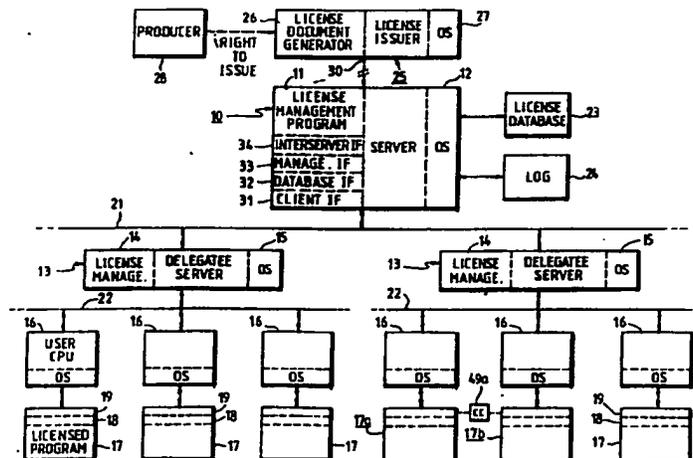
No protest accompanied the payment of additional search fees.



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁵ : G06F 1/00</p>	<p>A1</p>	<p>(11) International Publication Number: WO 92/20022 (43) International Publication Date: 12 November 1992 (12.11.92)</p>
<p>(21) International Application Number: PCT/US92/03812 (22) International Filing Date: 6 May 1992 (06.05.92) (30) Priority data: 697,652 8 May 1991 (08.05.91) US 723,456 28 June 1991 (28.06.91) US 722,840 28 June 1991 (28.06.91) US 723,457 28 June 1991 (28.06.91) US (71) Applicant: DIGITAL EQUIPMENT CORPORATION [US/US]; 146 Main Street, Maynard, MA 01754 (US). (72) Inventor: WYMAN, Robert, Mark; 410 Second Avenue, South No. 108, Kirkland, WA 98033 (US).</p>	<p>(74) Agents: NATH, Ram, B. et al.; c/o Joyce D. Lange, Digital Equipment Corporation, 111 Powdermill Road, Maynard, MA 10754 (US). (81) Designated States: AT, AT (European patent), AU, BB, BE (European patent), BF (OAPI patent), BG, BJ (OAPI patent), BR, CA, CF (OAPI patent), CG (OAPI patent), CH, CH (European patent), CI (OAPI patent), CM (OAPI patent), CS, DE, DE (European patent), DK, DK (European patent), ES, ES (European patent), FI, FR (European patent), GA (OAPI patent), GB, GB (European patent), GN (OAPI patent), GR (European patent), HU, IT (European patent), JP, KP, KR, LK, LU, LU (European patent), MC (European patent), MG, ML (OAPI patent), MR (OAPI patent), MW, NL, NL (European patent), NO, PL, RO, RU, SD, SE, SE (European patent), SN (OAPI patent), TD (OAPI patent), TG (OAPI patent). Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>	

(54) Title: MANAGEMENT INTERFACE AND FORMAT FOR LICENSE MANAGEMENT SYSTEM



(57) Abstract

A distributed computer system employs a license management system to account for software product usage. A management policy having a variety of alternative styles and contexts is provided. Each licensed product upon start-up makes a call to a license server to check on whether usage is permitted, and the license server checks a database of the licenses, called product use authorizations, that it administers. If the particular use requested is permitted, a grant is returned to the requesting user node. The product use authorization is structured to define a license management policy allowing a variety of license alternatives by values called "style", "context", "duration" and "usage requirements determination method". The license administration may be delegated by the license server to a subsection of the organization, by creating another license management facility duplicating the main facility. The license server must receive a license document (a product use authorization) from an issuer of licenses, where a license document generator is provided. A mechanism is provided for one user node to make a call to use a software product located on another user node; this is referred to as a "calling card", by which a user node obtains permission to make a procedure call to use a program on another node. A management interface allows a license manager at a server to modify the license documents in the database maintained by the server, within the restraints imposed by the license, to make delegations, assignments, etc. The license documents are maintained in a standard format referred to as a license document interchange format so the management system is portable and can be used by all adhering software vendors. A feature of the database management is the use of a filter function.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FI	Finland	MI	Mali
AU	Australia	FR	France	MN	Mongolia
BB	Barbados	GA	Gabon	MR	Mauritania
BE	Belgium	GB	United Kingdom	MW	Malawi
BF	Burkina Faso	GN	Guinea	NL	Netherlands
BG	Bulgaria	GR	Greece	NO	Norway
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IE	Ireland	RO	Romania
CA	Canada	IT	Italy	RU	Russian Federation
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland	KR	Republic of Korea	SN	Senegal
CI	Côte d'Ivoire	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	US	United States of America
DK	Denmark	MG	Madagascar		
ES	Spain				

MANAGEMENT INTERFACE AND FORMAT FOR LICENSE MANAGEMENT SYSTEM

BACKGROUND OF THE INVENTION

15 This invention relates to methods of operation of computer systems, and more particularly to a method and system for managing the licensing of software executed on computer systems.

20 In U.S. Patent 4,937,863, issued to Robert, Chase and Schafer and assigned to Digital Equipment Corporation, the assignee of this invention, a Software Licensing Management System is disclosed in which usage of licensed software may be monitored in a computer system to determine if a use is within the scope of a license. The system maintains a database of licenses for software products,

- 2 -

delivering the license document may be in the form of a network, or may be a phone line using modems, or may include physical delivery by disks or CD ROMs, for example. Likewise, the method of delivery of the software products being licensed, i.e., the applications programs 17 to be executed on the CPUs 16, is not material to the license management facility of the invention; the products are delivered by some appropriate means, e.g., the communications link 30 and the networks 21 and 22, by CD ROMs or disks physically distributed, etc.

Although shown in Figure 1 as operating on a distributed system, in the simplest case the license management facility of the invention may be operated on a single CPU. The license management program 11 and the applications program 17 may be executing on the same CPU, in which case the license document would be stored in a database 23 as before, on this CPU, and the calls from the unit 18 to the license server would be local instead of RPCs. As in the distributed system, however, the licensed product would still not have access to the license document, but instead could only make inquires to the server program, even if all are executing on the same CPU.

In operation of the distributed system of Figure 1, the producer 28 gives the issuer 25 authority to grant licenses on its behalf (the producer and issuer can be a single entity or multiple entities). The license document generator program 26, under control of a user (a person), generates a license (usually the result of negotiation between the user of program 26 and a user of the server 10). This license is called a product use authorization, and it is transmitted by the link 30 to the server 10. The license management program in the server 10 stores the product use authorization in the database 23, and, if delegation is an authorized option, may distribute parts of the authorized use to the delegatee servers 13,

- 3 -

where it is likewise stored in a database. Thereafter, administration of the license is only in response to inquiries from user nodes 16. When execution of a program 17 begins, the unit 18 is invoked to check on the availability of a license for this particular node. The unit 18 sends (as by an RPC) a request to the license management program 14 (or 11 if there is no delegatee), where the product use authorization stored in database 23 is checked to see if use is authorized. If so, a return is sent to the user node 16, granting permission to continue. When the program 17 has finished executing, the unit 18 again is invoked to signal to the license management program, again by an RPC, that the authorization is released, so the license management program can take appropriate action, e.g., log the use in log 24, etc.

To implement these operations, the license management program 11 or 14 contains several functions, including a client interface 31, a database interface 32, a management interface 33, and an interserver interface 34 for communicating with the delegates 13 (if any). The client interface 31, as described below, handles the requests received from the user nodes 16, and returns resulting from these requests. The database interface 32 handles the storing and retrieval of license information in the database 23, and logging license usage activity to log 24, and retrieval of this data. The management interface 33 handles the tasks of receiving the product use authorizations from the issuer 25 and maintaining the database 23 via the database interface 32. The interserver interface 34 handles the task of communicating with the delegatee servers 13, including transmitting the assigned parts of the product use authorizations, or communicating with other license servers that may be separately executing the license management function; for example, calls for validating calling cards may be made to another such server.

- 4 -

If there are no delegates or no other license servers, then of course the interserver interface 34 has no function, and is idle.

5 The license document or "product use authorization" forming the basis for the license management activity of the program 11 on the server 10 may be illustrated as a data structure containing the information set forth in Figure 2; in actual practice the product use authorization is preferably a more abstract data arrangement, not in such a rigidly structured format as illustrated. For example, the product use authorization as well as similar documents stored in the database 23, or passed between components of the system of Figure 1, may be of the so-called tag-length-value data format, where the data structure begins with an identifying tag (e.g., PUA or product use authorization) followed by a field giving the length, followed by the value itself (the content). One type of data treatment using this tag-length-value format is an international standard referred to as ASN.1 or Abstract Syntax Notation. In any event, the document 35 illustrated in Figure 2 is merely for discussing the various items of data, rather than representing the way the information is stored. Some of the fields shown here exist at some times and not others, and some are optional; the product use authorization may also include additional fields not shown or discussed here. Also it should be noted that copies of parts of this type of document are made for the delegates, so this representation of Figure 2 is a composite of several documents used in the system of Figure 1. The document 35 includes fields 36 identifying the software product by product name, producer, version numbers, release date, etc. The issuer 25 is identified in field 37, and the licensee (usually the owner of the license server 10) identified in field 38. The essential terms of the license grant are then defined in fields 40-46. The start date and end date are specified in fields 40; these store the exact time (date, hour, minute, second, etc.) when the license becomes valid and

10

15

20

25

- 5 -

when it ends, so licenses may be granted to start at some future time and to end at a particular time. Note that the previous practice has been to specify only the ending date, rather than also a start date as employed here. Each of the nodes, including issuer 25, servers 10 and 13, and user nodes 16, maintain a time value by a local clock referenced to a standard, so inherent in the license management facility is the maintaining of a time standard to compare with the start and end date information in the fields 40. The units granted are specified in field 41; the units are an arbitrary quantitative measure of program usage. In a delegatee server 13, the units field 41 will have some subset of the units field in the original product use authorization. As units are granted to users 16 or delegated, the remaining units available for grant are indicated in a subfield 42 in the copy of the document used by the server. The management policy occupies fields 43-46, and includes style, context, duration and LURDM (license use requirements determination method), as will be explained. The style field 43 specifies whether the licensed units are controlled by an "allocative" style or "consumptive" style, or some other "private" algorithm, where styles are ways used to account for the consumption or allocation of the units. The context field 44 specifies the location and environment in which product use or license management occurs, i.e., a CPU or an individual user or a network, etc. Duration field 45 indicates whether the license granted to a user is by assignment, by transaction, or immediate. The LURDM field 46 indicates the license use requirements determination method, in some cases using a license use requirements table (LURT) seen as field 47, as will be described.

Additional fields 48-54 in the product use authorization 35 of Figure 2 define features such as delegation authorization, calling authorization, overdraft

- 6 -

authorization, combination authorization, token, signature, checksum, etc. These will be described in the following paragraphs.

5 If the delegation field 48 is true, a license server 10 may distribute license units to multiple servers 13. A time limit may be imposed, i.e., units can be delegated to other hardware systems until they time out. Delegation allows an administrator to distribute units to improve response time and increase the resilience of the system. For example, the communication network 21 may include a satellite link to a remote facility where the local server 13 has a number of clients or users 16, in which case the calls to the server 13 would be completed
10 much quicker than would be the case if calls had to be made to the server 10. Also, delegation may be used as a method of allocating licensed units within a budget for administrative purposes. Usually the delegation authorization is a feature that is priced by the issuer, i.e., a license granting 1000 units with delegation authorization is priced higher than without this authorization.

15 The field 49 contains a calling authorization and/or a caller authorization. If the caller authorization in field 49 is true, the product is permitted to receive calls from other named products requesting use of the product, and if conditions are met (identified caller is authorized) the server can grant a calling card, as described below. If the calling authorization is true, the product can make calls
20 to other products. If neither is true, then the product can neither make or receive calls using the calling card feature. Referring to Figure 1, if product 17a wishes to make a remote procedure call to a feature of product 17b running on a different user node 16, it makes a call to its server 13 including a request for a calling card, and, if permitted, the return to product 17a includes a calling card
25 49a. The product 17a then makes a call to product 17b in the usual manner of

RPCs, sending along the calling card 49a, which the product 17b then verifies by a call to its server 13 before executing the called procedure and issuing its return to product 17a. The feature of calling cards is important for distributed applications. For example, if a product is able to execute faster in a distributed system by assigning tasks to other CPUs, then the issue is presented of which license policy is needed, i.e., does every node executing a part of the task have to be licensed and consume or receive allocation of a unit, or just the one managing the task? This is resolved for most applications by use of this calling card concept. The product use authorization for such a product has the calling authorization field 49 enabled, so calling cards can be issued. This feature is typically separately priced.

The combination authorization field 50 of Figure 2 determines whether or not license requests from a user node 16 can be satisfied by combining units from multiple product use authorizations. It may be advantageous to purchase licenses with different policy values, and use units from certain product use authorizations only for overflow or the like. Or, for other reasons, it may be advantageous to "borrow" and "lend" units among delegated servers or user nodes. This function is permitted or denied by the content of field 50.

The overdraft field 51 determines whether or not a requested allocation from a user node 16 will be nevertheless granted, even though the units available field 42 is zero or too small to permit the requested use. Overdrafts can be unlimited, or a specific overdraft pool can be set up by a server 10, for a customer's internal administrative purposes. That is, the overdraft value may be unlimited in the original license, but limited or zero for internally distributed copies of the license. Thus, the product use authorization sent by the issuer 25 to

5 the customer may have overdrafts permitted by the field 51, but the customer may deny overdraft permission for its own budgeting purposes. In any event, if overdraft is permitted, additional fees have to be paid to the issuer at some accounting period, when the logged usage from log 24 indicates the available units have been exceeded. If overdraft is denied, then the units 18 of the user nodes making request allocations are structured to inform the products 17 that a license grant is not available. The intent is not to prevent the application program from running; the license server merely informs the application whether or not the license manager determines that it is authorized to run. The application can itself be structured to shut itself down if not authorized to run, or it can be structured to shut down certain functions (e.g., ability to save files, ability to print, etc.), or it can be structured to continue in a fully functional manner. The purpose of the license management facility is not that of enforcement, nor that of "copy protection", but instead is merely that of license management.

15 An optional token field 52 is available in the product use authorization 35 of Figure 2. This field can contain comments or other information desired by the issuer or user. For example, a telephone support number may be included in the token field, then when the product 17 shows its "help screen" the number is inserted. This number would be part of the argument, i.e., data transmitted to the user node 16, when the server 10 makes a return following a request allocation message from the user. This field may also be used to store information used in a "private" style, where the information from this field returned to the user node is employed by the application program 17 or the stub 19 to determine if the application can be activated.

- 9 -

The signature field 53 in the product use authorization 35 is a part of a validation mechanism which provides important features. This field contains a digital signature encoded to reflect the data in the license itself, as well as other encoding methods not known to customers, so it cannot be duplicated unless the encoding algorithm is known. In a preferred embodiment, a so-called "public/private key" system of encoding is used for the signature field 53. The encoding algorithm used to generate the signature 53 is known to the issuer 25, using a private key, and anyone knowing the public key can decode the signature to determine if it is valid but cannot determine the encoding algorithm so it cannot produce a forged signature. So, if the server 10 knows the public key which is unique to the issuer 25, it can determine if a license document 35 is genuine, but it cannot itself generate license documents. However, if the server possesses a valid license document that gives it the right to delegate, then it will be assigned its own private key (different from all other issuers or servers) and its delegates 13 will be able to determine if a valid delegated license is delivered to them as they will be given the public key for the servers 13. The field 53 will thus contain both the original signature from the issuer 25 and the license server's signature when delivered to a delegatee 13. The decoding algorithm using a public key for any signatures is thus used by the license server 10 or delegatee 13 to make sure a product use authorization 35 is authentic before it is stored in the database 23. Related to the digital signature 53 is a checksum field 54, which merely encodes a value related by some known algorithm to the data in the product use authorization 35 itself. This field may be used merely to check for corruption of the data as it is stored, recalled, and transmitted within the system. That is, the checksum is used for data validation rather than security.

- 10 -

Two concepts central to the license management system implemented using the license document or product use authorization 35 of Figure 2 are the "license units", specified in field 41 or 42 and the "context", specified in field 44. License units are an abstract numerical measure of product use allowed by the license. When a product 17 (or a function or feature of a product) makes a license-checking request, the license management program 11 on server 10 computes how many license units are required to authorize this particular use of the product, and this is the license units requirement, in some cases using the LURDM field 46. A "context" is a set of tagged values which define the location and environment in which product use or license management occurs. Context values may be specified in field 44 of the product use authorization 35 of Figure 2 to restrict the environments in which the license may be managed and in which product use may occur. A context template may also be specified in the field 44 to indicate which parts of the complete context of product use (sub-contexts) are significant in differentiating product uses for the purposes of unit allocation; when this is specified, it allows separate product uses to share license units in a controlled way.

The two general types of policies specified in field 43 are allocative and consumptive. An allocative policy grants to the holder a specific number of license units (field 41) and specifies the policy which must be used to account for the allocation of these units. A software product 17 which is being managed by an allocative license will require verification that the appropriate number of license units have been allocated to it prior to performing services to the user. Typically, this allocation of units occurs either at the time of activation of the product 17 or at the time that product use is enabled on a particular platform (user CPU 16). The units typically remain allocated to the product 17 throughout the period that the product is running or is enabled to run. Upon termination of

- 11 -

processing or disabling, the allocated units are deallocated and made available for allocation to other instances of the software product 17 (other users 16 activating the product). In general, as long as any license units remain unallocated in field 42, the holder of the license is contractually authorized to increase his utilization of the licensed product. The usage does not deplete the license, however, as the units are returned to the units-available field 42 after a user is finished, and can be granted again to another user.

A consumptive unit based license, indicated in policy field 43, grants to the holder a specific number of initial license units (from field 42) and specifies the policy used to account for the consumption of those units. A software product 17 which is being managed by a consumptive license will cause an appropriate number of license units to be consumed to reflect the services provided by the product. Once consumed, units cannot be reused. Thus, the number of units available for future use declines upon every use of the licensed software product 17. This may also be referred to as a "metered" policy, being conceptually similar to measured consumption of electricity, water, etc. When the number of available units in field 42 reaches zero, the license may require that further use of the product is prohibited, or, the agreement may permit continued decrementing of the number of available units; the result is the accumulation of a negative number of available units in the field 42. It is anticipated that most consumptive unit based licenses will consider negative units to represent an obligation of the license holder to pay the license issuer 25. The transaction log 24 maintains an audit trail for providing a record of the units used in a consumptive license.

Referring to Figure 3, the major elements of the management policy are set forth in a table, where the possible entries for the fields 43, 44, 45 and 46 are

- 12 -

5 listed. For the style entry 43, the possibilities are allocative and consumptive as just described, plus a category called "private" which represents a style of management undefined at present but instead to be created especially for a given product, using its own unique algorithm. It is expected that most licenses may be administered using the named alternatives of Figure 3, but to allow for future expansion to include alternatives not presently envisioned, or to permit special circumstances for unique software, the "private" choices are included, which merely mean that the product 17 will generate its own conditions of use. It is important to note that, except for the "private" alternative, the license management is totally in control of the license management program 11 on the license server 10 (or delegatee 13), rather than at the product 17. All the product 17 does, via the unit 18, is to make the request inquiry to the server 10 via the client interface 31, and report when finished.

15 The context field 44 specifies those components (sub-contexts) of the execution-context name which should be used in determining if unit allocations are required. License data is always used or allocated within, or for the benefit of, some named licensing context, and context can include "platform contexts" and "application contexts". Platform contexts are such things as a specific network, an execution domain, a login domain, a node, a process ID or a process family, a user name, a product name, an operating system, a specific hardware platform, as listed in Figure 3. Applications contexts are information supplied from the application (the product 17), such as may be used in a "private" method of determining license availability. The context name can use several of these, in which case the context name is constructed by concatenating the values of all subcontexts into a single context name, e.g., a VAX 3100 platform using VMS operating system.

20

25

The duration field 45 defines the duration of an allocation of license units to a specific context or the duration of the period which defines a valid consumptive use. For durations of type "Assignment," the specification of a reassignment constraint is also provided for, as discussed below. There are three types of duration, these being "transaction," "assignment" and "immediate" as seen in Figure 3.

The transaction duration type, when specified for an allocative policy, indicates that license units should be allocated to the specified context upon receipt of a license request and that those units should be deallocated and returned to the pool of available units upon receipt of a corresponding license release from a user node 16. Abnormal termination of the process or context having made the original license request will be semantically equivalent to a license release. On the other hand, when specified for a consumptive policy, this duration type indicates that license units should be allocated to the specified context upon receipt of a license request and permanently removed from the available units pool (field 42) upon receipt of a license release which reflects successful completion of the transaction. Upon receipt of a license release which carries an error status or upon abnormal termination of the processor context having made the original license request, the allocated units will be deallocated and returned to the pool of available units (field 42).

The assignment duration type in Figure 3 (field 45 of Figure 2) imposes the constraint that the required units must have been previously assigned to a specific context. The sub-contexts which must be specified in the assignment are those given in the context-template. A "reassignment constraint" may be imposed, and this is a limitation on how soon a reassignment can be made. For example, a

reassignment constraint of 30-days would require that units assigned to a specific context could not be reassigned more often than every 30-days; this would prevent skirting the intent of the license by merely reassigning units whenever a user of another context made a request allocation call for the product. Related to this assignment constraint, a "reallocation limit" may also be imposed, to state the
5 minimum duration of an allocation; where there is a context template of process, the intent is to count the number of uses of the software product at a given time, but where software runs in batch rather than interactive mode it may run very quickly on a powerful machine, so a very few concurrent uses may permit almost
10 unlimited usage - by imposing a reallocation constraint of some time period, this manner of skirting the intent of the license may be constrained.

The immediate duration type (field 45 of Figure 2) is used to indicate that the allocation or consumption of an appropriate number of license units from the pool of available units (field 42) should be performed immediately upon receipt
15 of a license request. Receipt of license release or abnormal terminations will then have no impact on the license management system. When specified as the duration for an allocative policy, the effect will be simply to check if an appropriate number of license units are available at the time of a license request. When specified as the duration for a consumptive policy, the effect will be to
20 deduct the appropriate number of license units from the available pool at the time of a license request, and, thereafter, abnormal termination, such as a fault at the user CPU 16 or failure of the network link, will not reinstate the units.

The LURDM or license unit requirement determination method, field 46, has the alternatives seen in Figure 3 and stores information used in calculating the
25 number of units that should be allocated or consumed in response to a license

- 15 -

request. If this field specifies a table lookup kind, this means license unit requirements are to be determined by lookup in the LURT (field 47) which is associated with the current license. If a constant kind is specified, this indicates that the license units requirements are constant for all contexts on which the licensed product or product feature may run. A private LURDM specifies that the license unit requirements are to be determined by the licensed product 17, not by the license management facility 11. The license unit requirements tables (LURTs) provide a means by which issuers of licenses can store information describing the relation between context (or row selector) and unit requirements. The license units requirements determination method (LURDM) must specify "table lookup" for the LURT to be used, and if so a row selector must be specified, where a valid row selector is any subcontext, e.g., platform ID, user name, time of day, etc. An example of an LURT fragment is shown in Figure 4, illustrating the license unit requirements table mechanism. In this example, the row selector is "platform-ID" so the platform-ID value determines which row is used. The issuer of this LURT of Figure 4 has established three unit requirement tiers for use in determining the unit requirements for that issuer's products. The reason for the tiers is not mandated by the license management system, but the issuer 25 (actually the user of the program 26) would probably be establishing three pricing tiers, each reflecting a different perspective on the relative utility of different platforms in supporting the use of various classes of product 17. The first column in Figure 4, Column A, specifies the use requirements for a class of products whose utility is highly sensitive to the characteristics of the specific platform on which they are run. This can be seen by observing that the unit requirements are different for every row in Column A. Products which use the second column (Column B) appear to have a utility which is more related to the class of platform on which they run. This is indicated by the fact that all the PC

- 16 -

platforms share a single value which is different from that assigned to the VAX platform. The final column (Column C) is for use with a class of products which is only supported on the VAX platform. Figure 4 is of course merely an example, and the actual LURT created by the license document generator 26 and stored in the license database 23 (as field 47 of the product use authorization 35) can be of any content of this general format, as desired by the license issuer.

Instead of always selecting the rows in LURT tables according to the platform ID of the execution platform, in order to handle the breadth of business practices that need to be supported by the license management facility, the LURT mechanism is extended by providing a "row selector" attribute in the LURT class structure. No default is provided although it is expected that the normal value for the row selector attribute will be "platform ID."

In the system of patent 4,937,863, a concept similar to that of the LURT of Figure 4 was provided, with rows selected by the platform ID and columns selected by some arbitrary means, typically according to product type. The system of this invention allows flexibility in the selection of both LURT row and column while continuing to provide backwards compatibility for licenses defined within the constraints of patent 4,937,863.

Some examples will illustrate potential uses for the row selector attribute. A customer may only want to pay for the use of a product during one or two months of the year; the product may be FORTRAN and the reason for this request may be that the company has a fairly stable set of FORTRAN subroutines that are given regular "annual maintenance" only during the months of May and June. To handle this customer's needs, the FORTRAN product would generate

- 17 -

an application subcontext which would contain a value representing the month of the year. Then, a LURT table would be defined with twelve rows, one for each month of the year. In some column, probably column A, a negative one (-1) would be placed in each month except for May and June. These two months would contain some positive number. The product use authorization would then have a LURDM field specifying a LURT for use to determine the units requirement, and would name this custom LURT table. The effect would be that the PUA could only be used during the months of May and June since negative one is interpreted by license managers to mean "use not authorized." This mechanism could also be used to do "time of day" charging. Perhaps charging fewer units per use at night than during the day. Also, if a subcontext was used that contained a year value, a type of license would be provided that varied in its unit requirements as time passed. For instance, it might start by costing 10-units per use in 1991 but then cost one unit less every year as time passed, eventually getting to the point where the unit requirement was zero.

Another example is font names. A specific customer may purchase a license giving it the right to concurrent use of 100-units of a large font collection; some of the fonts may cost more to use than others. For instance, Times Roman might cost 10-units per use while New Century Schoolbook costs 20-units per use. The problem is, of course, making sure that charges are properly made. The solution is to build a LURT table with a specified application subcontext as its row selector. A row is then created for each font in the collection and in Column A of the LURT, the number of units required to pay for use of the font would be specified. The print server would then specify the name of a font as the value of the application subcontext whenever it does an *lm_request_allocation()* call. This will allow charges to be varied according to font name.

5 A further example is memory size. Some products are more or less valuable depending on the size of memory available to support them. A software vendor wishing to determine unit requirements based on memory size will be able to do so by building LURT tables with rows for each reasonable increment of memory (probably 1-megabyte increments). Their applications would then sense memory size (using some mechanism not part of the license management facility) and pass a rounded memory size value to the license manager in a private context.

10 Other examples are environment and operating system. Some products may be valued differently depending on whether they are being run in an interactive mode or in batch. This can be accomplished by building LURT rows for each of the standard platform subcontexts that specify environment. Regarding operating system, it has been considered desirable by many to have a single product use authorization permit the use of a product on any number of operating systems, this conflicts with some vendors policies who do not want to have to create a single price for a product that applies to all operating systems. 15 Thus, if an operating system independent license were offered for a C compiler, the price would be the same on MS-DOS, VMS, and/or UNIX. Clearly, it can be argued that the value of many products is, in part, dependent on the operating system that supports them. By using a row selector of operating system (one of the standard platform subcontexts), license designers could, in fact, require 20 different numbers of units for each operating system. However, it might be more desirable to base the row selection on a private application subcontext that normally had the same value as the operating system subcontext. The reason for this is that the license designer might want to provide a default value for operating system names that were unknown at the time the LURT rows were defined. If 25 this is the case, the product would contain a list of known operating systems and

- 19 -

pass the subcontext value of "Unknown" when appropriate. The LURT row for "Unknown" would either contain a negative one (-1) to indicate that this operating system was unsupported or it would contain some default unit requirement.

5 Another example is variable pricing within a group. One of the problems with a "group" license is that there is only one unit requirements field on the PUA for a group. Thus, all members of the group share a single unit requirement. However, in those cases where all members of the group can be appropriately licensed with a constant unit requirement yet it is desired to charge different amounts for the use of each group member, a LURT can be built that has rows defined for each group member. The row selector for such a group would be the standard platform subcontext "product name."

10

Many different types of license can be created using different combinations of contexts, duration and policy from the table of Figure 3. As examples, the following paragraphs show some traditional licensing styles which can be implemented using the appropriate values of the product use authorization fields 43-46.

15

A "system license" as it is traditionally designated is a license which allows unlimited use of a product on a single hardware system. The correct number of units must be allocated to the processor in advance and then an unlimited product use is available to users of the system. The product use authorization would have in the context field 44 a context template for a node name, the duration field would be "assignment" and the policy style field 43 would be "allocative".

20

5 A "concurrent use" license is one that limits the number of simultaneous uses of a licensed product. Concurrent use license units are only allocated when the product is being used and each simultaneous user of the licensed product requires their own units. In this case the context template, field 44, is a process ID, the duration field is "transaction" and the policy style 43 is "allocative".

10 A "personal use" license is one that limits the number of named users of a licensed product. This style of licensing guarantees the members of a list of users access to a product. Associated with a personal use type of product use authorization there is a list of registered users. The administrator is able to assign these users as required up to the limit imposed by the product use authorization; the number of units assigned to each user is indicated by the LURDM. It may be a constant or it may vary as specified in a LURT. The context template is "user name", the duration is "assignment", and the policy is "allocative".

15 A "site license" is one that limits the use of a licensed product to a physical site. Here the product use authorization contains for the context template either "network name" or "domain name", the duration is "assignment" and the policy style field 43 is "allocative".

20 Generally, a license to use a software product is priced according to how much benefit can be gained from using the product, which is related to the capacity of the machine it will run on. A license for unlimited use on a large platform such as a mainframe, where there could be thousands of potential users at terminals, would be priced at a high level. Here the style would be "allocative", the context template = "node", the duration = "assignment" and the LURDM may be "Column A" - the units, however, would be large, e.g., 1000. At the other end

- 21 -

of the scale would be a license for use on a single personal computer, where the field values would be the same as for the mainframe except the units would be "1". If a customer wanted to make the product available on the mainframe but yet limit the cost, he could perhaps get a license that would allow only five users at any given time to use the product; here the fields in the product use authorization would be: units = 5; style = allocative; context template = process; duration = transaction; LURDM = constant, 1-unit. This would still be priced fairly high since a large number of users may actually use the product if a session of use was short. A lower price would probably be available for a personal use license where only five named persons could use the product, these being identified only in the license server 10, not named by the license issuer 25. Here the fields in the product use authorization are: units = 5; style = allocative; context template = user name; duration = transaction; LURDM = constant, 1-unit.

An additional feature that may be provided for in the product use authorization 35 is license combination. Where there are multiple authorizations for a product, license checking requests sent by user nodes 16 may be satisfied by combining units from multiple authorizations. Individual product use authorizations may prohibit combined use. Thus, a licensee may have a license to use a product 17 on an allocative basis for a certain number of units and on a consumptive basis for another number of units (this may be attractive from pricing standpoint); there might not be enough units available for a particular context from one of these licenses, so some units may be "borrowed" from the other license (product use authorization), in which case a combination is made.

The interface between the program executing on the client or user 16 and the license server 10 or its delegates 13 includes basically three procedure calls:

- 22 -

a request allocation, a release allocation and a query allocation. Figure 5 illustrates in flow chart form some of the events occurring in this client interface. The request allocation is the basic license checking function, a procedure call invoked when a software product 17 is being instantiated, functioning to request an allocation of license units, with the return being a grant or refusal to grant. Note that a product may use request allocation calls at a number of points in executing a program, rather than only upon start-up; for example, a request allocation may be sent when making use of some particular feature such a special graphics package or the like. The release allocation call is invoked when the user no longer needs the allocation, e.g., the task is finished, and this return is often merely an acknowledge; if the style is consumptive, the caller has the opportunity via the release allocation call to influence the number of units consumed, e.g., decrease the number due to some event. The query allocation call is invoked by the user to obtain information about an existing allocation, or to obtain a calling card, as will be described.

The request allocation, referred to as *lm_request_allocation()*, is a request that license units be allocated to the current context. This function returns a grant or denial status that can be used by the application programmer to decide whether to permit use of the product or product feature. The status is based on the existence of an appropriate product use authorization and any license management policies which may be associated with that product use authorization. License units will be allocated or consumed, if available, according to the policy statement found on the appropriate product use authorization. The product would normally call this function before use of a licensed product or product feature. The function will not cause the product's execution to be terminated should the request fail. The decision of what to do in case of failure to obtain allocation of license

- 23 -

units is up to the programmer. The arguments in a request allocation call are the product name, producer name, version, release date, and request extension. The product name, producer name, version and release date are the name of the software product, name of producer, version number and release date for specifically identifying the product which the user is requesting an allocation be made. The request extension argument is an object describing extended attributes of the request, such as units required, LURT column, private context, and comment. The results sent back to the calling node are a return code, indicating whether the function succeeded and, if not, why not, and a grant handle, returned if the function completes successfully, giving an identifying handle for this grant so it can be referred to in a subsequent release allocation call or query allocation call, for example.

The release allocation, referred to as *lm_release_allocation()*, is an indication from a user to the license manager to release or consume units previously allocated. This function releases an allocation grant made in response to a prior call to request allocation. Upon release, the license management style 38 determines whether the units should be returned to the pool of available units or consumed. If the caller had specified a request extension on the earlier call to request allocation which contained a units-required-attribute, and the number of units requested at that time are not the number of units that should be consumed for the completed operation, the caller should state with the units-consumed argument how many units should be consumed. The arguments of the release allocation are: grant handle, units consumed, and comment. The grant handle identifies the allocation grant created by a previous call to request allocation. The units-consumed argument identifies the number of units which should be consumed if the license policy is consumptive; this argument should only be used

in combination with an earlier call to request allocation which specified a units requirement in a request extension. Omission of this argument indicates that the number of units to be consumed is the same as the number allocated previously. The comment argument is a comment which will be written to the log file 24 if release units are from a consumptive style license or if logging is enabled. The result is a return code indicating if the function succeeded, and, if not, why not.

The query allocation, or *lm_query_allocation()*, is used by licensed products which have received allocations by a previous request allocation call. The query is to obtain information from the server 10 or delegatee server 13 about the nature of the grant that has been made to the user and the license data used in making the grant, or to obtain a calling card (i.e., a request that a calling card be issued). Typically, the item read by this query function is the token field 52 which contains arbitrary information encoded by the license issuer and which may be interpreted as required by the stub 19 for the licensed product software 17, usually when a "private" allocation style or context is being employed. The arguments in this procedure call are the grant handle, and the subject. The grant handle identifies the allocation grant created by a previous call to request allocation. The subject argument is either "product use authorization" or "calling card request"; if the former then the result will contain a public copy of the product use authorization. If this argument is a calling card request and a calling card which matches the previous constraints specified in that request can be made available, the result will contain a calling card. If the subject argument is omitted, the result will contain an instance of the allocation. The results of the query allocation call are (1) a return code, indicating whether the function succeeded, and, if not, why not, and (2) a result, which is either an allocation, a product use authorization or a calling card, depending on type and presence of the subject argument.

- 25 -

Referring to Figure 5, the flow chart shows the actions at the client in its interface with the server. When the software product 17 is to be invoked, the unit 18 is first executed as indicated by the block 60, and the first action is to make a request allocation call via the stub 19, indicated by the block 61. The client waits
5 for a return, indicated by the loop 62, and when a return is received it is checked to see if it is a grant, at decision block 63. If not, the error code in the return is checked at block 64, and if a return code indicates a retry is possible, block 65, control passes back to the beginning, but if no retry is to be made then execution is terminated. If the policy is to allow use of the product 17 without a license
10 grant, this function is separately accounted for. If the decision point 63 indicates a grant was made, the grant handle is stored, block 66, for later reference. The program 17 is then entered for the main activities intended by the user. During this execution of product 17, or before or after, a query allocation call can be made, block 67, though this is optional and in most cases not needed. When
15 execution of the program 17 is completed, the grant handle is retrieved, block 68, and a release allocation call is made, block 69. A loop 70 indicates waiting for the return from the server, and when the return received it is checked for an error code as before, and a retry may be appropriate. If the release is successfully acknowledged, the program exits.

20 Referring to Figure 6, the actions of the server 10 or delegatee server 13 in executing the license management program 11 or 14, for the client interface, are illustrated in flow diagram form. A loop is shown where the server program is checking for receipt of a request, release or query call from its clients. The call would be a remote procedure call as discussed above, and would be a message
25 communicated by a network, for example. This loop shows the decision blocks 71, 72 and 73. If a release allocation call is received, a list of products for which

authorizations are stored is scanned, block 74, and compared to the product identity given in the argument of the received call, block 75. If there is no match, an error code is returned to the client, block 76, and control goes back to the initial loop. If the product is found, the authorization is retrieved from the database 23, block 77 (there may be more than one authorization for a given product, in which case all would be retrieved, but only one will be referred to here) and all of the information is matched and the calculations made depending upon the management policy of Figures 3 and 4, indicated by the decision block 78. If a grant can be made, it is returned as indicated at block 79, or if not an error code is returned, block 80. If a release allocation call is received, indicated by a positive at the decision block 72, the grant handle in the argument is checked for validity at block 81. If no match is found, an error code is returned, block 82, and control passes back to the initial loop. If the handle is valid, the authorization for this product is retrieved from the database 23 at block 83, and updated as indicated by the block 84. For example, if the license management style is allocative, the units are returned to the available pool. Or, in some cases, no update is needed. The authorization is stored again in the database, block 85, and a return made to the client, block 86, before control passes back to the initial loop. If the decision block 73 indicates that a query allocation call is received, again the grant handle is checked at block 87, and an error code returned at block 88 if not valid. If the grant handle matches, the authorization is retrieved from the database 23, at block 89, and a return is made to the client giving the requested information in the argument, block 90.

The basic allocation algorithm used in the embodiment of the license management system herein described, and implemented in the method of Figures 5 and 6, is very simple and can handle a very large proportion of known license

unit allocation problems. However, it should be recognized that a more elaborate and expanded algorithm could be incorporated. Additions could be made in efforts to extend the allocation algorithm so that it would have specific support for optimizing unit allocation in a wider variety of situations. Particularly, sources of non-optimal allocations occurring when using the basic allocation algorithm are those that arise from combination and reservation handling.

The first step is formation of full context. The client stub 19 is responsible for collecting all specified platform and application subcontexts from the execution environment of the product 17 and forwarding these collected subcontexts to the license management server 13 or 10. The collection of subcontexts is referred to as the "full context" for a particular license unit allocation request.

The next step is retrieval of the context template. When the license manager receives an *lm_request_allocation()*, it will look in its list of available product use authorizations (PUA) to determine if any of them conform to the product identifier provided in the *lm_request_allocation()* call. The product identifier is composed of: product name, producer, version, release date. If any match is found, the license manager will extract from the matching PUA the context template. This template is composed of a list of subcontexts that are relevant to the process of determining unit requirements. Thus, a context template may indicate that the node-ID subcontext of a specific full context is of interest for the purposes of unit allocation. The context template would not specify any specific value for the node-ID; rather, it simply says that node-ID should be used in making the allocation computation.

The next step is masking the full context. Having retrieved the context template, the license manager will then construct an "allocation context" by filtering the full context to remove all subcontexts which are not listed in the context template. This allocation context is the context to be used in determining allocation requirements.

5

Then follows the step of determining if the request is new. The license manager maintains for each product use authorization a dynamic table which includes the allocation contexts of all outstanding allocations for that PUA (i.e., allocations that have been granted but have not yet been released). Associated with each entry in this table is some bookkeeping information which records the number of units allocated, the full context, etc. To determine if a recent *lm_request_allocation()* requires an allocation of units to be made, the license manager compares the new allocation context with all those allocation contexts in the table of outstanding allocations and determines if an allocation has already been made to the allocation context. If the new allocation context does not already exist in the table, an attempt will be made to allocate the appropriate number of units depending on the values contained in the LURDM structure of the PUA and any LURTs that might be required. If an allocation context similar to that specified in the new allocation request does exist in the table, the license manager will verify that the number of units previously allocated are equal to or greater than the number of units which would need to be allocated to satisfy the new allocation request. If so, the license manager will return a grant handle to the application which indicates that the allocation has been made (i.e., it is a "shared allocation" - the allocated units are shared between two requests.) If not, the license manager will attempt to allocate a number of units equal to the

10

15

20

25

difference between the number previously allocated and the number of units required.

5 The step of releasing allocations (Fig. 6, blocks 84-85) occurs when the license manager receives an *lm_release_allocation()* call; it will remove the record in its dynamic allocation table that corresponds to the allocation to be released. Having done this, the license manager will then determine if the allocation to be removed is being shared by any other allocation context. If so, the units associated with the allocation being released will not be released. They will remain allocated to the remaining allocation contexts. Some of the units might
10 be released if the license manager determines that the number of allocated units exceeds the number needed to satisfy the outstanding allocation contexts. If this is the case, the license manager will "trim" the number of allocated units to an appropriate level.

15 In summary, the two things that make this algorithm work are (1) the basic rule that no more than one allocation will be made to any single allocation context, and (2) the use of the context template to make otherwise dissimilar full contexts appear to be similar for the purposes of allocation.

20 The license designer's task, when defining basic policy, is then to determine which contexts should appear to be the same to the license manager. If the license designer decides that all contexts on a single node should look the same (context template = node-ID), then any requests that come from that node will all share allocations. On the other hand, a decision that all contexts should be unique (i.e., context template = process-ID) will mean that allocations are never shared.

and stores a unit value indicating the number of licensing units for each product. When a user wishes to use a licensed product, a message is sent to the central license management facility requesting a license grant. In response to this message, the facility accesses the database to see if a license exists for this product, and, if so, whether units may be allocated to the user, depending upon the user's characteristics, such as the configuration of the platform (CPU) which will execute the software product. If the license management facility determines that a license can be granted, it sends a message to the user giving permission to proceed with activation of the product. If not, the message denies permission.

While the concepts disclosed in the patent 4,937,863 are widely applicable, and indeed are employed in the present invention, there are additional functions and alternatives that are needed in some applications. For example, the license management system should allow for simultaneous use of a wide variety of different licensing alternatives, instead of being rigidly structured to permit only one or only a few. When negotiating licenses with users, vendors should have available a wide variety of terms and conditions, even though a given vendor may decide to narrow the selection down to a small number. For example, a software product may be licensed to a single individual for use on a single CPU, or to an organization for use by anyone on a network, or for use by any users at terminals in a cluster, or only for calls from another specific licensed product, or any of a large number of other alternatives. A vendor may have a large number of products, some sold under one type of license and some under others, or a product may be a composite of a number of features from one or more vendors having different license policies and prices; it would be preferable to use the same license management system for all such products.

5 Distributed computing systems present additional licensing issues. A distributed system includes a number of processor nodes tied together in a network of servers and clients. Each node is a processor which may execute programs locally, and may also execute programs or features (subparts of programs) via the network. A program executing on one node may make remote procedure calls to procedures or programs on other nodes. In this case, some provision need be made for defining a license permitting a program to be executed in a distributed manner rather than separately on a single CPU, short of granting a license for execution on all nodes of a network.

10 In a large organization such as a company or government agency having various departments and divisions, geographically dispersed, a software license policy is difficult to administer and enforce, and also likely to be more costly, if individual licenses are negotiated, granted and administered by the units of the organization. A preferred arrangement would be to obtain a single license from
15 the software producer, and then split this license into locally-administered parts by delegation. The delays caused by network communication can thus be minimized, as well as budgetary constraints imposed on the divisions or departments. Aside from this issue of delegation, the license management facility may best be operated on a network, where the licensing of products run on all
20 nodes of the network may be centrally administered. A network is not necessary for use of the features of the invention however, since the license management can be implemented on a single platform.

25 Software products are increasingly fragmented into specific functions, and separate distribution of the functions can be unduly expensive. For example, a spreadsheet program may have separate modules for advanced color graphics, for

accessing a database, for printing or displaying an expanded list of fonts, etc. Customers of the basic spreadsheet product may want some, none or all of these added features. Yet, it would be advantageous to distribute the entire combination as one package, then allow the customer to license the features separately, in various combinations, or under differing terms. The customer may have an entire department of the company needing to use the spreadsheet every day, but only a few people who need to use the graphics a few days a month. It is advantageous, therefore, to provide alternatives for varied licensing of parts or features of software packages, rather than a fixed policy for the whole package.

Another example of distribution of products in their entirety, but licensing in parts, would be that of delivering CD ROMs to a customer containing all of the software that is available for a system, then licensing only those parts the customer needs or wishes to pay fees for rights to use. Of course, the product need not be merely applications programs, operating systems, or traditional executable code, but instead could also include static objects such as printer fonts, for example, or graphics images, or even music or other sound effects.

As will be explained below, calling and caller authorizations are provided in the system according to one feature of the invention, in order to provide technological support for a number of business practices and solve technical problems which require the use of what is called "transitive licensing." By "transitive licensing" is meant that the right to use one product or feature implies a right to use one or more other products or features. Transitive licenses are similar to group licenses in that both types of license consist of a single instrument providing rights of use for a plurality of products. However, transitive licenses differ from group licenses in that they restrict the granted rights by specifying that

the licensed products can only be used together and by further specifying one or more permitted inter-product calling/caller relationships. Some examples may help to clarify the use and nature of a transitive license: the examples to be explained are (1) two products sold together, (2) a give-away that results from narrow choices of licensing alternatives, (3) a client licensing method in a client/server environment, (4) impact of modular design, and (5) the impact of distributed design.

A software vendor might have two products for sale: the first a mail system, and the second a LEXISTM-like content-based text retrieval system. Each of these products might be valued at \$500 if purchased separately. Some customers would be satisfied by purchasing the rights to use only one of these products. others might find that they can justify use of both. In order to increase the likelihood that customers will, in fact, purchase both products, it would not be surprising if the software vendor offered his potential customers a volume discount, offering the two products for a combined price of \$800. The customers who took advantage of this combined offer would find that they had received two products, each of which could be exploited to its fullest capabilities independently from the other. Thus, these customers would be able to use the content based retrieval system to store and retrieve non-mail documents. However, from time to time, the vendor may discover that particularly heavy users of mail wish to be able to use the content based retrieval system only to augment the filing capabilities provided by the standard mail offering. It is likely that many of these potential customers would feel that \$800 is simply too much to pay for an extended mail capability. The vendor might then consider offering these customers a license that grants mail users the right to use the content-based retrieval system only when they are using mail and prohibits the use of content

based retrieval with any other application that might be available on the customers system. This type of license is referred to below a "transitive license," and it might sell for \$600.

5 Another example is a relational database product (such as that referred to as Rdb™) designed for use on a particular operating system, e.g., VMS. This relational database product has two components: (1) A user interface used in developing new databases, and (2) a "run-time" system which supports the use of previously developed databases. The developers of the database product might spend quite a bit of effort trying to get other products made by the vendor of the database product to use it as a database instead of having those other products
10 build their own product-specific databases. Unfortunately, the other product designers may complain that the cost of a run-time license for the database product, when added to the cost of licenses for their products, would inevitably make their products uncompetitive. Thus, some mechanism would be needed that would allow one or another of the vendor's products to use the run-time system for the relational database product in a "private" manner while not giving
15 unlicensed access to products of other vendors. No such mechanism existed, prior to this invention; thus, the vendor might be forced to sell the right to use its run-time system for the database product with its proprietary operating system license. Clearly, this combined license would make it possible for the vendor's products to
20 use its database product without increasing their prices; however, it also would make it possible for any customers and third-parties to use the database product without paying additional license fees. However, had the system of the invention been available, the vendor could have granted transitive licenses for the run-time
25 component of its database product to all the vendor's products. Essentially, these licenses would have said that the database run-time could be used without an

- 35 -

additional license fee if and only if it was used in conjunction with some other of the vendor's products. Any customer wishing to build a new relational database application or use a third-party application that relied on the vendor's database product would have had to pay the vendor for its database run-time license.

5 A proposed client/server licensing method provides yet another example of a problem which could be solved by transitive licensing. Typically, a client is only used by one user at a time, while a server can support an arbitrary number of clients depending on the level of client activity and the capacity of the machine which is supporting the server. While traditionally, server/client applications have
10 been licensed according to the number of clients that a server could potentially support, this may not be the most appropriate method for licensing when the alternatives afforded by the invention are considered. The business model for the proposed client/server method requires that each client be individually licensed and no explicit licensing of servers is required to support properly licensed clients.
15 Such a licensing scheme makes it possible to charge customers only for the specific number of clients they purchase. Additionally, it means that a single client can make use of more than one server without increasing the total cost of the system. The solution to this transitive licensing problem would be to provide a mechanism that would allow the clients to obtain license unit allocations and then pass a
20 "proof" of that allocation to any servers they may wish to use. Servers would then support any clients whose proofs could be verified to be valid. On the other hand, if a client that had not received a proof of allocation attempted to use a server, the server would obtain a license allocation for that client session prior to performing any services. Such a solution has not been heretofore available.

- 36 -

As the complexity and size of the software systems provided to customers increases, it is found that the actual solution provided to customers is no longer a single product. Rather, customers are more often now offered solutions which are built up by integrating an increasing number of components or products, each of which can often stand alone or can be part of a large number of other solutions. In fact, a product strategy may rely almost exclusively on the vendor's engineering and selling a broad range of specialized components that can only be fully exploited when combined together with other components into a larger system. Such components include the relational database runtime system mentioned above, mail transport mechanisms, hyperinformation databases, document format conversion services, time services, etc. Because these components are not sold on their own merits, but rather on their ability to contribute to some larger system, it is unlikely that any one customer will be receiving the full abstract economic value of any one of the components once integrated into a system. Similarly, it can be observed that the value of any component once integrated into a larger system varies greatly from system to system. Thus, it may be found that a mail transport mechanism contributes a large part of a system whose primary focus is mail, however, it will contribute proportionally less of the value of a system that provides a broader office automation capability. As a result of these observations, the job of the business analyst who is attempting to find the "correct" market price for each component standing on its own, is more complex. In reality, the price or value of the component can only be determined when considering the contribution of that component to the full system or solution in which it is integrated. Attempting to sell the components at prices based on their abstract, independent values will simply result in overpriced systems.

- 37 -

Transitive license styles are particularly suited to dealing with pricing of modular components, since component prices can be clearly defined in relation to the other components or systems which they support. Thus, a vendor can charge a price of \$100 for the right to use a mail transport system in conjunction with one product, yet charge \$200 for the use of the same mail transport system when used by another product.

In addition to the "business" reasons for wanting to support transitive licensing, there is also a very good technical reason that arises from the growing tendency of developers to build "distributed products" as well as the drive toward application designs that exploit either tightly or loosely coupled multiprocessor systems; the availability and growing use of remote procedure calls has contributed to this tendency. This technical problem can be seen to arise when considering a product which has a number of components, each of which may run in a different process space and potentially on a different computer system. Thus, there might be a mail system whose user interface runs on one machine, its "file cabinet" is supported by a second machine and its mail transport system runs on yet a third machine. The simple question which arises is: "Which of the three components should check for licenses?" Clearly it must be ensured that no single component can be used if a valid license is not present. Thus, the answer to the question will probably be that all three components should check for licenses. However, the question is then presented: "Where are the licenses to be located?". This can become more complex.

Increasingly, the distributed systems being built are being designed so that it is difficult to predict on which precise machine any particular component will run. Ideally, networks are supposed to optimize the placement of functions

5 automatically so that the machine with the most available resource is always the one that services any particular request. This dynamic method of configuring the distribution of function servers on the network makes it very difficult for a system or network manager to predict which machines will run any particular function and thus very difficult for him to decide on which machines software licenses should be loaded.

10 Even if a system manager could predict which machines would be running the various application components and thus where the license units should be loaded, the situation would still be less than ideal. The problem arises from the fact that each of the components of the application would be independently making requests for license unit allocations. This behavior will result in a difficult problem for anyone trying to decide how many license units are required to support any one product. Given the mail example, the problem wouldn't exist if it were assumed that all three components (i.e., user interface, file cabinet, and transport system) were required by the design of the mail system to be in use simultaneously. If this were the case, it could be simply assumed that supporting a single activation of the mail system would require three units. However, in a real mail system, it will be inevitably discovered that many users will only be using just the user-interface and file-cabinet components of the system at one time. Thus, there will be some unused units available which could be used to authorize additional users. This situation might not be what is desired by the software vendor.

25 The problem of providing license support to multi-component products which are dynamically configured could be solved by viewing each of the product components as a distinct licensable product and by treating the problem as one

of transitive licensing, but a mechanism for accomplishing this has not been available. Essentially, a single license document would be created that stated that if any one of the components had successfully obtained a license to run, it could use this grant to give it the right to exploit the other components. Thus, in the
5 example above, the user might start the mail system by invoking its user interface. This user interface code would then query the license management facility for a license allocation and once it has received that allocation, it would pass a proof of allocation to the other mail components that it uses. Each of the other components would request that the license management system validate that the
10 "proof" is valid prior to performing any service; however, none of the other components would actually require specific allocations to be made to them. In this way, the complexity of licensing and managing networks of distributed applications can be significantly reduced.

SUMMARY OF THE INVENTION

15 In accordance with one embodiment of the invention, a license management system is used to account for software product usage in a computer system. The system employs a license management method which establishes a management policy having a variety of simultaneously-available alternative styles and contexts. A license server administers the license, and each licensed product
20 upon start-up makes a call to the license server to check on whether usage is permitted, in a manner similar to that of patent 4,937,863. The license server maintains a store of the licenses, called product use authorizations, that it administers. Upon receiving a call from a user, the license server checks the product use authorization to determine if the particular use requested is

permitted, and, if so, returns a grant to the requesting user node. The license server maintains a database of product use authorizations for the licensed products, and accesses this database for updating and when a request is received from a user. While this license management system is perhaps of most utility on a distributed computer system using a local area network, it is also operable in a stand-alone or cluster type of system. In a distributed system, a license server executes on a server node and the products for which licenses are administered are on client nodes. However, the license management functions and the licensed products may be executing on the same processor in some embodiments.

The product use authorization is structured to define a license management policy allowing a variety of license alternatives by components called "style", "context", "duration" and "usage requirements determination method". The style may be allocative or consumptive. An allocative style means the units of the license may be allocated temporarily to a user when a request is received, then returned to the pool when the user is finished, so the units may be reused when another user makes a request. A consumptive style means the units are deducted from an available pool when a user node makes a valid request, and "consumed", not to be returned for reuse. The context value defines the context in which the use is to be allowed, such as on a particular network, by a particular type of CPU, by a particular user name, by a particular process, etc. The duration value (used in conjunction with the style component) concerns the time when the license units are to be deducted from the available pool of units, whether at the time of request, after a use is completed, etc. A usage requirements determination method may be specified to define or provide information concerning the number of license units charged in response to a license request from a user node; for example, some CPU platforms may be charged a larger number of license units

than others. A table may be maintained of usage requirements, and the determination method may specify how to access the table, for example. The important point is that the user node (thus the software product) can only make a request, identifying itself by user, platform, process, etc., and the license management facility calculates whether or not the license can be granted (that is, units are available for allocation), without the user node having access to any of the license data or calculation. There is a central facility, the license server, storing the license documents, and, upon request, telling the licensed products whether they can operate under the license terms.

An important feature of one embodiment is that the license administration may be delegated to a subsection of the organization, by creating another license management facility duplicating the main facility. For example, some of the units granted in the product use authorization may be delegated to another server, where the user nodes serviced by this server make requests and receive grants.

The license management facility cannot create a license itself, but instead must receive a license document (a product use authorization) from an issuer of licenses. As part of the overall license management system of the invention, a license document generator is provided which creates the product use authorizations under authority of the owner of the software, as negotiated with customers. Thus, there are three distinct rights in the overall license management facility of the invention: (1) the right to issue licenses, (2) the right to manage licenses, and (3) the right to use the licensed products. Each one of these uses the license document only in prescribed ways. The license issuer can generate a license document. The license manager (or license server as referred to herein) can grant products the right to use under the license, and can delegate parts of the

- 42 -

licensed units for management by another server, as defined by the license document; the way of granting rights to products is by responding to certain defined calls from the products. And, the licensed products can make certain calls to the license server to obtain grants of rights based upon the license document, inquire, or report, but ordinarily cannot access the document itself.

As explained above, transitive licensing is an important feature of one embodiment. This is the provision of a mechanism for one user node to get permission to use another software product located on another user node; this is referred to as a calling authorization and a caller authorization, using a "calling card," and these are examples of the optional features which must be specifically permitted by the product use authorization. A user node must obtain permission to make a procedure call to use a program on another node; this permission is obtained by a request to the license server as before, and the permission takes the form of a calling card. When a calling card is received by a second node (i.e., when the procedure call is made), a request is made by the second node to the license server to verify (via the product use authorization) that the calling card is valid, and a grant sent to the user node if allowed. In this manner, all nodes may have use of a program by remote calls, but only one consumes license units.

Another important feature of one embodiment is a management interface which allows a license manager to modify the license policy components of a license document maintained by at a license server in its database. Usually the license manager can only make modifications that restrict the license policy components to be more restrictive than originally granted. Of course, the management interface is used to make delegations and assignments, if these are authorized.

The license document interchange format is an important feature, in that it allows the license management system to be used with a wide variety of software products from different vendors, so long as all follow the defined format. The format uses data structures that are defined by international standards.

5 An important function is the filter function, used in the management interface and also in the client interface to select among elements in the data structures.

BRIEF DESCRIPTION OF THE DRAWINGS

10 The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as other features and advantages thereof, will be best understood by reference to the detailed description of specific embodiments which follows, when read in conjunction with the accompanying drawings, wherein:

15 Figure 1 is a diagram in block form of a distributed computer system which may be used to implement the license management operations according to one embodiment of the invention;

Figure 2 is a diagram of the content of a license document or "product use authorization" generated by the license document generator and stored by the license server in the system of Figure 1;

Figure 3 is a diagram of the alternatives for license style, context and duration making up the license management policy implemented in the system of Figure 1, according to one embodiment of the invention;

5 Figure 4 is a diagram of an example of a fragment of a license use requirements table (LURT) used in the system of Figure 1, according to one embodiment of the invention;

Figure 5 is a logic flow chart of a program executed by a user node (client), in the system of Figure 1, according to one embodiment of the invention;

10 Figure 6 is a logic flow chart of a program executed by a license server, in the system of Figure 1, according to one embodiment of the invention; and

Figure 7 is a diagram of the calls and returns made in an example of use of calling cards in the system of Figure 1.

Figure 8 is a diagram of an LDIF document identifier, according to an standard format;

15 Figure 9 is a syntax diagram of an LDIF document;

Figure 10 is a diagram of an LDIF document structure;

Figures 11, 13, 15, 17, 18, 19, 21-28 and 31-43 are syntax diagrams for elements of various ones of the LDIF data structures;

Figure 16 is a diagram of a license data structure;

Figures 12, 14 and 20 are examples of descriptions of data elements using a standard notation;

5 Figures 29 and 30 are examples of context templates used in the license management system;

Figures 44 and 45 are tables of attributes specific to filter and filter item type; and

Figure 46 is notation in a standard format for an example of a filter.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

10 Referring to Figure 1, a license management facility according to one example embodiment of the invention is centered around a license server 10, which typically includes a CPU located in the customer's main office and executing a license management program 11 as will be described, under an operating system 12. The license server 10 communicates with a number of delegates 13 which
15 likewise include CPUs in departments or divisions of the company or organization, each also executing a license management program 14 under an operating system 15. The license management program 14 is the same as the program 11 executing on the main server 10; the only difference in the functions of server 10 and servers 13 is that the latter have a delegated subset of the license units granted to the
20 server 10, as will be described. The CPUs 13 are in turn servers for a number of

users 16, which are CPU nodes where the licensed programs 17 are actually executed. The programs 17 executing on the user CPUs 16 are applications programs (or operating systems, etc.) which have added to them units 18 and 19, according to the invention, allowing them to make inquiry to the their server 13 (or 10) before executing and to report back after executing, using a client stub 19 in the manner of remote procedure calls, in one embodiment. A user node 16 may have many different programs 17 that may be executed, and the various user nodes 16 would usually each have a set of programs 17 different from the other user nodes, all of which would be administered by the license management program 14 or 11. The terms "program" and "licensed product" are used in reference to the element 17, but it is understood that the products being administered may be segments of programs, or functions or features called by another program, or even merely data (such as printer fonts), as well as complete stand-alone applications programs. The license server 10 communicates with the delegatee servers 13 by a network 21, as is usual in large organizations, and the delegatee servers 13 each communicate with their user nodes 16 by networks 22; these networks may be of the Ethernet, token ring, FDDI types or the like, or alternatively, the user nodes 16 may be merely a cluster of terminals on a multiuser system with the delegatee being a host CPU. The particular hardware construction of the user nodes, server nodes, communication networks, etc., and the operating systems 12 or 15, are of no concern regarding the utility of the features of the invention, the only important point being that the user CPUs 16 of the software products 17 in question can communicate readily and quickly with their respective server nodes 13 or 10. In one embodiment, remote procedure calls (RPCs) are used as the communication medium for the interfaces between components of the system, handling the inquiries and grants as will be described.

A remote procedure call is similar to a local procedure call but is made to a procedure located on a remote node, by way of a communications network.

5 The function of the unit 19 is that of a client stub, in a remote procedure call sense. The calls to the license server 10 are made through this stub 19, and returns are received by the stub 19 and passed on to the program 17. The stub 19 is responsible for obtaining the network addresses of other nodes on the network, such as the server 10. Also, the stub 19 is responsible for determining the context (as defined below) for passing on to the server 10. The unit 18 functions to execute a "private" type of license availability determination if this is used, rather than this task being done by the application program 17, but if the ordinary method of determination is employed (using the license server) as is usually the case, the unit 18 is merely code that starts the execution and passes calls and returns back and forth between the program 17 and the unit 19.

15 The license server 10, using the license management program 11, maintains a license data file 23 comprising a number of license documents or licenses (product use authorizations), and also maintains a log 24 which is a record of the usage activity of all of the user CPUs 16 of each of the licensed programs. The delegatee servers 13 would maintain similar license databases and logs. The license server 10 has no authority to originate a license, but instead must receive a license from a license issuer 25. The issuer 25 is again a CPU executing a license document generator program 26 under an operating system 27. The license issuer 25 may be under control of the producer 28 of the programs or software products being licensed, or may be controlled by a distributor who has received the authority to grant licenses from the producer or owner 28. The communications link 30 between the license issuer 25 and the license server 10 for

5 This mechanism permits the system of the invention to dispose of the cumbersome, explicit support of license types having different scope such as the cluster licenses, node licenses, and process licenses found in prior license management systems including that of patent 4,937,863. Instead of defining a limited set of scopes (cluster, node, etc.), the system of this invention provides a general mechanism which allows an effectively unlimited range of allocation scopes to be defined.

10 Transitive licensing, as referred to above, is supported by the system of the invention by (1) calling authorizations, which are statements made in field 49 of the product use authorization 35 for one product (the "caller") to permit that product to call another product (the "callee"), and, (2) caller authorizations, which are statements made in field 49 of the product use authorization for one product (the "callee") to permit it to be called by another product (the "caller").

15 If calling or caller authorizations are to be exploited by products, then whenever one product calls another product, it must pass the callee a calling card 49a. This calling card 49a is an encoding of an identification of the caller as well as a statement by the license management system that a license unit allocation has been made to the caller which is passing the calling card. This calling card is then passed by the callee to the license management system for validation and, if the
20 either the product use authorization of the caller carries an appropriate calling authorization or the product use authorization of the callee carries an appropriate caller authorization, the use of the callee by the caller will be authorized without requiring any additional license unit allocations.

Referring to Figure 7, the intercomponent interactions that occur when either calling or caller authorizations are being used are illustrated. This figure shows a license management server 10, a caller product 17a named "Product-1" and a callee product 17b named "Product-2". When Product-1 starts to run, it will make an *lm_request_allocation()* call to the license management server 10 to obtain a grant handle for an allocation of some number of units of the Product-1 license. Either immediately, or at some later time, but always prior to making a call to Product-2, Product-1 will call *lm_query_allocation()*, passing the grant handle received earlier and specifying that it wants a calling card for the product named "Product-2." If the field 49 of the product use authorization 35 used to satisfy the grant represented by the grant handle carries a calling authorization in field 49 naming "Product-2," the license manager will create a calling card 49a which includes the statement that a calling authorization exists and pass this calling card back to Product-1. If the calling authorization does not exist, the calling card passed to Product-1 will contain a statement to that effect.

Once Product-1 has successfully obtained a calling card 49a from the license manager, it will then make a call to Product-2, passing the calling card along with any other initialization parameters that would normally be used when starting Product-2. Product-2 will then pass that calling card to the license manager as part of its *lm_request_allocation()* call and the license manager will determine if the calling card is valid. Note that calling cards become invalid once the process which received the calling card makes an *lm_release_allocation()* call or terminates abnormally. If the calling card is valid, and it indicates that a calling authorization is present, the license manager will verify this statement and if found to be true, will return a grant handle to Product-2. If, on the other hand, the calling card carries an indication that no calling authorization is present, the

license manager will attempt to find a product use authorization for Product-2 that contains a caller authorization naming Product-1 as an authorized caller. If the caller authorization is found, a grant handle will be passed back to Product-2. If not, the license manager will ignore the calling card and proceed with the normal
5 *lm_request_allocation()* logic.

The requirement to be passing calling cards between products requires that both the caller and the callee be "aware" of the fact that calling and caller authorizations may be used. This is one of the few examples of a requirement for a product 17 to become actively involved in the licensing problem when using the
10 licensing management system of the invention. However, since the use of calling/caller authorizations is a fairly "sophisticated" and powerful feature, it is considered acceptable to impose this burden on application coders.

MANAGEMENT INTERFACE

Referring to Figure 1, the license management program 11 executing on a
15 server 10 includes a license management interface 33 which functions to allow a user at a console for the server 10 CPU or at a remote terminal to implement certain necessary operations. The management interface 33 is essentially the tools or mechanisms available to the license manager at the licensee's site to (a) load the various licenses received from issuers 25 into the database 23 and make them
20 available for request allocation calls from the users, (b) remove the licenses from the machine when expired, (c) to make delegations if permitted, (d) to make assignments, (e) to make reservations, etc. Whatever the license manager is allowed to do to modify the license for his special circumstances (within the

original grant, of course), he does it by the mechanism of the management interface 33. Some licenses are not modified at all, but merely loaded. In a multiple machine environment, as on a network, there is considerable modification, as it is necessary to make sure the correct number of units are distributed onto the correct machines, the right people have access, other people don't have access, etc. Thus, in a network environment, there is extensive use of the management interface 33.

In reference to the terminology used in describing the management interface, as well as the license management system in general, it is helpful to note that the documentation conventions, data declarations, macro declarations, etc., for the object management used in one embodiment of the invention are according to the standards set forth in *OSI Object Management API Specification, Version 2.0*, X.400 API Association and X/Open Company Limited, 24 August 1990, a published document.

The specific operations available to the management interface 33 are to allow a manager to open and close a management session, register (load) objects in the license database 23, obtain a list of objects in the license database 23, and control a cursor (a cursor is a movable pointer to a member of a list of items). Once an object in the license database 23 is identified with the cursor, certain changes may be made in the object by a write function. For example, certain fields of a license document of Figure 2 or an LURT of Figure 4 may be changed in only specified ways as will be explained.

The operation of opening a session goes by the name of *lm_open_session()* and is used to establish a license management service session between a

- 52 -

management client and the service. Opening a session also creates a workspace to contain objects returned as a result of functions invoked within the session. Object management Objects can be created and manipulated within this workspace. Objects created within this workspace, and only such objects, may be used as Object arguments to the other license management service management functions used during the session established by a call to this function. More than one session may exist simultaneously.

The arguments that go with a *lm_open_session()* call are (a) the binding handle, which is binding information that defines one possible binding (a client-server relationship), and (b) a comment which will be inserted in the log file if logging is enabled. The results from a *lm_open_session()* call are (a) a return code indicating whether the function succeeded, and, if not, why not, (b) a session, which is an established license management session between the management client and the license management service, and (c) a workspace that will contain all objects returned as a result of functions invoked in the session.

The close session call is referred to by *lm_close_session()* and functions to terminate the lm session. This function terminates the license service management session and makes the argument unavailable for use with other interface functions. The arguments that go with a *lm_close_session()* call are (a) the session which identifies the established lm session between the management client and the license management service, and (b) a comment which will be inserted in the log file if logging is enabled. The result of the call is a return code indicating whether the function succeeded, and, if not, why not.

- 53 -

The list function returns a set of selected objects in the license database 23, and uses the name *lm_list_licenses()*. This function is used to search the license database 23 and return a cursor which represents the first of one or more objects which match the specified filter. The specified filter will be applied to each object in the license database 23; all objects for which the filter evaluates true will be included in the object list accessible by the *set_cursor* function. The arguments that go with *lm_list_licenses()* are (a) session which identifies an established session between the management client and the license management service, and (b) a filter which is an object used to select license database 23 objects; license database objects will only be included in the object list headed by the cursor if they satisfy the filter - the constant no-filter may be used as the value of this argument if all license data objects are to be included in the object list. The results of the *lm_list_licenses()* call are (a) a return code indicating whether the function succeeded, and, if not, why not, and (b) a license list upon successful completion of this call containing a cursor which represents the first of one or more objects in the current license database 23 for which the specified filter evaluates true.

The register function is to register objects in the license database 23, and uses the name *lm_register()*. This function is used to register (i.e., load or create) new objects, or modify existing objects, in the license database 23; the objects which may be registered include only those which are subclasses of the license data class or history objects. The arguments are (a) session, which identifies an established session between the management client and the license management service, (b) license data object which is to be registered; if this argument is omitted, the comment argument is a required argument and a history object containing the comment will be registered in the license database 23, and (c)

comment, which will be inserted in the log file if logging is enabled. The result is a return code indicating whether the function succeeded, and, if not, why not. The errors possible when it does not succeed include data-expired, duplicate-object, no-such-session, memory-insufficient, network-error, etc., indicated by this return code.

The set cursor function establishes a new cursor, and is called by *lm_set_cursor()*. The arguments are (a) session, which identifies an established session between the management client and the license management service, (b) forward, which is a boolean value indicating if the direction in which the cursor is to be moved is forward or reverse, (c) filter which is used to eliminate cursors from the search for the next cursor that are not wanted; a new cursor will only be set if it satisfies the filter - the constant no-filter may be used as the value of this argument if any cursor is to be considered as the target cursor, and (d) the cursor which is to be used as the starting point in searching for the new cursor. The results are (a) a return code indicating whether the function succeeded, and, if not, why not, and (b) next-cursor, which is the requested cursor. The error codes in the return code may be end-of-list, not-a-cursor, etc.

After a session is opened, and an object such as a product use authorization or a LURT has been identified by the cursor, using the functions explained above, the management interface 33 is able to execute certain object management interface functions such as write or copy. By this mechanism, the management interface can modify certain limited attributes. None of these attributes can be modified in such a way that they reduce constraints established by corresponding attributes in the license data objects. The more important attributes which can be modified by the management interface 33 using this mechanism are:

- 55 -

(a) assignment: an assignment of some or all of the units granted on the associated product use authorization;

(b) reservation: a reservation of some or all of the units granted on the associated product use authorization;

5 (c) delegation: a delegation of the right to manage some or all of the units granted on the associated product use authorization, or if the associated license data is not a product use authorization, the delegation is of the right to use that license data;

10 (d) backup delegation: a statement of the right to manage some or all or the units granted on the associated product use authorization; this right is only active at times when the delegating server is not available;

(e) allocation: an allocation of units to a specific context;

15 (f) allocation period: the minimum duration of a single allocation - all allocated units cannot be allocated to a new context until a time period equal to the allocation period has passed since the units were last allocated;

(g) termination date: a date which is to override the value specified as the end date of the product use authorization 40 - this date must be earlier than specified;

20 (h) delegation permitted: an override of the delegation permitted flag of the associated license data;

(i) overdraft: the current overdraft level;

(j) overdraft logging: an override of the overdraft logging attribute of the associated product use authorization;

25 (k) comment: a comment created by the licensee;

(l) extended info: information not defined by the architecture which may be of use in managing the license data.

5 It will be noted that an assignment and a reservation are identical, the only difference being that a reservation is something optional, while an assignment is something that is required. If the duration is Assignment in the policy declaration of Figure 3, the license manager must assign some or all of the units before units can be allocated. Reservations, on the other hand, are made by the license manager using the management interface, regardless of the policy.

10 Thus, there are certain attributes that can be changed by a license administrator using the management interface at the server 10, but none of these can result in obtaining more extensive rights to use than granted by the product use authorization. In each case, the license administrator can limit the rights which will be allocated to users in some way that may be appropriate for the administrator for control purposes.

LICENSE DOCUMENT INTERCHANGE FORMAT

15 The major structural components of an ASN.1 encoded document which conforms to the specifications for the license management system discussed above will be described. The object identifier that is assigned to this data syntax, according to one embodiment, is that specified in ASN.1 as seen in Figure 8. The International Standards Organization or ISO, as it is referred to, defines how bit patterns are chosen to uniquely identify an object type, so the bit pattern set forth
20 in Figure 8 would precede each document used in the license management system so the document could be identified as being a document conforming to the prescribed License Document Interchange Format.

5 A document encoded according to this format is represented by a value of a complex data type called "license document interchange format document" of LDIFDocument, in this embodiment. A value of this data type represents a single document. This self-describing data structure is of the syntax defined in the international standard ASN.1 referred to above. The X/Open standard referred to above defines the conventions that must be used in employing this syntax, while the syntax itself is described in an OSI (Open Systems Interconnect, a standard administered by ISO) document identified as X.409 (referenced in the X/Open document identified herein).

10 The LDIFDocument data type consists of an ordered sequence of three elements: the document descriptor, the document header, and the document itself. Each of these elements are in turn composed of other elements. The overall structure of the LDIFDocument data type will be described, and the nature of the document descriptor and document header types. Then, the document content
15 elements will be described in detail, as well as the various component data types used in the definition of the descriptor, the header and the content.

20 The LDIFDocument represents a single license document, with the syntax being shown in Figure 9 and the high-level structure of an LDIF document in graphical form being seen in Figure 10. The DocumentDescriptor of Figure 9 is a description of the document encoding, the DocumentHeader contains parameters and processing instructions that apply to the document as a whole, and the DocumentContent is the content of the document, all as explained below.

Referring to Figure 9, what this says is that an LDIFDocument is composed of (::= means "is composed of") a number of elements, the first thing in an

LDIFDocument is a bit pattern (tag) according to an international standard, indicating a certain type of document follows, which is indicated here to be "private" or vendor selected, the number 16373 in this case. Following the bit pattern which functions as a "starting delimiter" it is "implicit" that a "sequence" of elements must follow, where a sequence is distinguished from a set. A sequence is one or more of the elements to follow, whereas a set is exactly one of the elements to be listed. Implicit means that any file identified as LDIFDocument must have a sequence data type, rather than some other type. In the case of Figure 9, the sequence is document-descriptor, document header and document content; the document-content is mandatory, whereas the first two are optional. If an element in the sequence begins with a "0" it is a document-descriptor, "1" means a document-header, and "2" means it is a document-content. Again, it is implicit that the data following is of the format DocumentDescriptor, etc., in each case, and these are defined in Figure 11, Figure 13 and Figure 15.

Each file is in the tag-length-value format mentioned above, and also each element of a file containing multiple elements is of the tag-length-value format. The data stream could be examined beginning at any point, and its content determined by first looking for a tag, which will tell what data structure this is, then a length field will say how long it is, then the content will appear. These structures are nested within one another; a document containing several product-use-authorizations would be an LDIFDocument of the format of Figure 9, with a number of DocumentContent elements of Figure 15 following, with the length given for the LDIFDocument spanning the several PUAs, and the length given for each PUA being for the one PUA.

In Figure 11, the elements major-version and minor-version are seen to be "implicit integer". This means that because the element is of the type major-version, etc.. it must be an integer. Various other implicit types are given in other syntax diagrams, such as character-string, boolean, etc.

5 In Figure 15, the license body is identified as being of the type "choice" meaning it can be one of PUA, LURT, GroupDefinition, KeyRegistration, etc. Thus, knowing this is a license-body does not mean the data type of the object is known; it is a bit further where the kind of a license-body becomes known. The definition of a license body is not implicit, but instead is a choice type.

10 The contents of the various data elements will now be described in detail with reference to Figures 11-43. Using these detailed descriptions, the exact format of each of the elements used in the LDIF can be interpreted.

15 The license document descriptor or DocumentDescriptor consists of an ordered sequence of four elements which specify the version level of the LDIF encoding and identify the software that encoded the document, with the syntax being shown in Figure 11. An example of the way a product called PAKGEN V1.0 is expressed in the DocumentDescriptor encoding is shown in Figure 12. The fields in the DocumentDescriptor syntax are major-version, minor-version, encoder-identifier and encoder-name. The major-version field is the primary
20 indicator of compatibility between LDIF processors and the encoding of the present document; this major-version field is updated if changes are made to the system encoding that are not backward compatible. The minor-version field is the revision number of the system encoding. The encoder-identifier field is a registered facility mnemonic representing the software that encoded the document;

the encoder-identifier can be an acronym or abbreviation for the encoder name -
this identifier is constant across versions of the encoder. The encoder-identifier
should be used as a prefix to Named Value Tags in Named Value Lists to identify
the encoder of the named value. The encoder-name field is the name of the
5 product that encoded the document; the encoder-name string must contain the
version number of the product.

The document header or **DocumentHeader** contains data that pertains to
the document as a whole, describing the document to processors that receive it;
the syntax is shown in Figure 13. An example of a document header is shown in
10 Figure 14, using the hypothetical product **PAKGEN V1.0** of Figure 12. The
private-header-data contains the global information about the document that is not
currently standardized; all interpretations of this information are subject only to
private agreements between parties concerned, so a processor which does not
understand private header data may ignore that data. The **Title** field is the user-
15 visible name of the document. The **Author** field is the name of the person or
persons responsible for the information content of the document. The **Version**
field is the character string used to distinguish this version of the document from
all other versions. The **Date** field is the date associated with this document. Note
that the nature and significance of the **Title**, **Author**, **Version**, and **Date** fields can
20 vary between processing systems.

The content of an LDIF document is represented by a value of a complex
data type called **DocumentContent**. An element of this type contains one or more
LicenseData content element using a syntax as shown in Figure 15. There are no
restrictions on the number, ordering or context of **LicenseData** elements. The
25 structure of a **LicenseData** element is represented in Figure 16. No restrictions

- 61 -

are made on the number, ordering, or context of LicenseData elements. The license-data-header field of Figure 16 specifies that data, common to all types of license data, which describes the parties to the licensing agreement, the term of the agreement, and any constraints that may have been placed on the management of the license data encoded in the license body. The license-body is an element that contains one content element, including: product use authorizations, license unit requirements tables, group definitions, key registrations, and various forms of delegations. The Management-Info is an element that contains information concerning the current state of the license data; this element is not encoded by Issuers.

The license data header, called LicenseDataHeader, is represented as a syntax diagram in Figure 17. The license-id field provides a potentially unique identification of the encoded license data, so issuers of license data can generate unique license-ids to distinguish each issuance of license data; however, the architecture does not require this to be the case, since the only architectural restriction is that no two objects in any single license management domain may have the same value for license-id. The licensee field identifies the party who has received the rights reflected in the license data; there are at least two parties involved in all transfers of license data, first, the issuer of the license data, and second, the licensee or recipient of that data - it is anticipated that individual licensees will specify to those issuing them licenses what the licensee fields on their license data should contain. the term field identifies the term during which the license data may be used; the validity of license data can be limited by issuers to specific time ranges with given starting and ending dates, which are carried in the term element - attempts to use license data or products described by that data either before the start date or after the end date will result in conforming license

- 62 -

managers denying access to the license. Management-constraints identifies constraints placed on the right to manage the associated license data; these constraints can include (a) limiting the set of contexts permitted to manage the data, (b) limiting the set of platforms which may benefit from that management, and (c) limiting the right to backup and delegate the managed data. The signature provides the digital signature used by the issuer to sign the license data and identifies the algorithm used in encoding the signature. Issuer-comment is a comment provided by the issuer and associated with the license data.

The IssuerComment is of an informational nature and does not impact the process of authorizing product or feature use. This field is not included in the fields used to generate the signature for a license, thus, even if specified by an issuer, the IssuerComment can be omitted from a license without invalidating the license. If specified, the IssuerComment should be stored in the appropriate license data base with the associated license data. The IssuerComment can be retrieved by products which use the system and may be of particular utility to products in the "Software Asset Management" domain which are intended to extend or augment the administrative or accounting facilities or basic system components. Some examples of potential uses for this field are order information, additional terms and conditions, and support information. For order information, some issuers may wish to include with their loadable license data some indication of the purchase order or orders which caused the license data to be issued; licensees may find it useful to include this data in their license databases to assist in the license management process. For additional terms and conditions, the system will never provide automatic means for the management of all possible license terms and conditions, and so some issuers may wish to include summaries of non-system managed terms and conditions in the comment as a reminder. For

support information, the IssuerComment could be used to record the phone numbers or addresses of the responsible individuals within the issuing organization who should be contacted if there are problems with the data as issued.

5 A product use authorization as previously discussed in reference to Figure 2 is used to express the issuance of a right to use some product, product feature, or members of some product group. As such, it records the identity of the product for which use is authorized and specifies the means that will be used by the license manager to ensure that the licensee's actual use conforms to the terms and conditions of the license. Figure 18 illustrates a syntax diagram for a
10 ProductUseAuthorization. Product-id identifies the name of the producer of the product or product feature of which usage rights are being granted as well as the name of that product; in addition, issuers of product use authorizations may specify a range of versions and/or releases whose use is controlled by the specific product use authorization. Units-granted - Contains the number of units of
15 product use which are granted by the license. Management-policy defines the policy which is to be used in managing the granted software usage rights; this definition specifies the Style, Context-Template, Duration, and License Unit Requirements Determination Method which must be used. The calling-authorizations and caller-authorizations are as explained above in reference to
20 calling cards. The execution-constraints field identifies constraints placed on the characteristics of execution contexts which may be authorized to benefit from the units granted by this Product Use Authorization. The product-token field contains product specific data not interpreted in any way by any processors conformant with the architecture; software product producers 28 use this array to augment the
25 capabilities of conformant license managers.

5 Some anticipated uses of the token field include language support, detailed feature authorizations, and product support number. For language support, a token could be constructed which contains a list of local language interface versions whose use is authorized; thus, if a product were available in English, German, French and Spanish, a token could be constructed listing only English and German as the authorized languages. For detailed feature authorizations, some license issuers will wish to have very fine control over the use of features in a complex product; however, they may not wish to issue a large number of individual Product Use Authorizations to "turn on" each feature, so these vendors could construct tokens which contain lists of the features authorized or whose use is denied. For product support number, some issuers may wish to include on the product use authorization, and thus make available to the running product, some information concerning the support procedures for the product; for example, an issuer might include the telephone number of the support center or a support contract number, and the product could be designed to retrieve this data from the license manager and display it as part of Help dialogues.

20 The LURTs or license use requirements tables of Figure 4 provide a means by which issuers of licenses, whose LURDM is dependent on the type of platform on which the product is run, can store information describing the relationship between the platform type and unit requirements. A syntax diagram for a LURT is shown in Figure 19. In Figure 20, an example of how the LURT of Figure 4 might be encoded is illustrated. Lurt-name specifies the name by which the LURT is to be known to conforming license managers. The rows field models a list of multicolumn lurt rows. Platform-id identifies the platform for which this LurtRow provides license unit requirements. The lurt-columns field provides a list of one or more lurt column values; the first value provided is

- 65 -

assigned to column-1 of the lurt-row, the second value provided is assigned to column-, etc. A lurt column value of -1 indicates that use of the product or feature is not authorized, while a lurt column value of 0 or greater indicates the number of units that must be allocated in order to authorize product use on the platform described by this lurt-row. All unspecified columns (e.g., columns whose number is greater than the number of column values provided in the lurt columns element) will be considered to contain the value -1.

In reference to Figure 19, to use the row-selector feature mentioned above, the platform-ID element would be replaced with *row-selector* which would be implicit of Context. Also, in Figure 34 described below, in the lurdm-kind element, *row-selector* would be included if the row-select feature is to be used.

As discussed above, Figure 4 provides an example of a hypothetical LURT, illustrating the LURT mechanism, where the issuer of this LURT table has established three unit requirement tiers for use in determining the unit requirements for that issuer's products. Figure 20 provides an example of how the LURT presented in Figure 4 might be encoded.

A group definition is used to define and name a license group. Once so defined, the name of this group can be used on product use authorizations in the same manner as a product name. Since a single product use authorization specifies the management policy for all members of the group, the members of that group must be compatible in their licensing styles, i.e., a personal use type product can not be mixed with a concurrent use product in the same group. Figure 21 shows a group definition syntax diagram. Group-name is the name which must appear on Product Use Authorizations for this group. Group-version

- 66 -

5 specifies the current version of this group; the requirements for matching between the version information on a product use authorization and that on a specified group definition are the same as those rules which require matching between produce use authorizations and the Release Date data provided by products. Group-members lists those products or features which are components of the named group.

10 A key registration is used by a producer 28 or issuer 25 who have been registered as authorized license issuers and provided with an appropriate public and private key pair. The key registration identifies the public key which is to be used by conforming license managers 10 in evaluating signatures 53 created by the named issuer 25 or producer 28. A key registration syntax diagram is shown in Figure 22. Key-owner-name provides the name which must be used in either of, or both, of the Producer and Issuer fields of license data generated by the issuer; the key-owner-name must be identical to that specified in the Issuer field of the header record. Key-algorithm identifies the registered algorithm that is to be used 15 when producing digital signatures with this key. Key-value identifies the public key.

20 An issuer delegation is typically issued by a producer 28 and authorizes the named issuer 25 to issue licenses for products produced by the producer. An issuer delegation syntax diagram is shown in Figure 23. Delegated-issuer-name identifies the name which must appear in the Issuer field of any Product Use Authorization generated using the License Issuer Delegation. Delegated-product-id identifies the products whose licenses the named issuer is authorized to issue. Delegated-units-granted, if specified, indicates that the use of this IssuerDelegation is to be managed in the style of a consumptive license; the value of this attribute 25

- 67 -

gives the number of units for which license documents may be generated (i.e., if granted 1000 units by a Producer, an Issuer can only issue 1000 units.) Template-authorization provides a "template" Product Use Authorization whose attribute values must be included on any Product Use Authorization generated using this IssuerDelegation; in the case of attributes which have a scalar value (i.e., Version, Release Date, etc.), the Issuer may issue licenses with more restrictive values than those specified on the Template Authorization. Sub-license-permitted indicates whether the Issuer identified on this IssuerDelegation may issue an IssuerDelegation for the delegated-product-id.

A license delegation, as shown in a syntax diagram of Figure 24, is used to delegate the right to manage license data. Such delegations are created by the licensee (by the license manager), if authorized by the issuer 28. A backup delegation, also shown in Figure 24, is used by one license management facility to authorize another to manage the delegated rights in the case that the delegating license manager is not running. The delegated-units field specifies the number of units whose management is being delegated; this may only be specified when a product use authorization is being delegated. Delegation-distribution-control defines the mechanisms by which the distribution and refreshing of the delegation will be accomplished. Delegatee-execution-constraints identifies any constraints which are placed on the execution-context of the Delegatee; these constraints are applied in addition to those which are a part of the delegated License Data. Assignment-list identifies any assignments of the delegated units that must be respected by the delegatee. Delegated-data stores a copy of the LicenseData received from the issuer that is the subject of the delegation; the delegated data is not provided when the LicenseDelegation element is included in a DelegationList.

- 68 -

The management information or ManagementInfo element records information concerning the current state of the LicenseData with which it is associated. A syntax diagram of the ManagementInfo element is shown in Figure 25. The assignments field identifies a list of one or more assignments which may be outstanding for the units on the associated product use authorization.

5 Reservations identifies a list of one or more reservations which may be outstanding for the units on the associated product use authorization. Delegations identifies a list of all outstanding delegations. Backup-delegations identifies all outstanding backup delegations. the allocations field provides detailed

10 information about outstanding allocations which involve units from the associated product use authorization. Registration-date is the date on which the LicenseData was registered in the license database. Registrar is the context which caused the LicenseData to be registered. Local-comment is a comment field. Termination-

15 date means a license defined date after which the license data may not be used; this date must be earlier than the end-date specified in the license data's term record. The extended-info field allows additional information concerning the state of the LicenseData and its handling by the license manager that is not standardized.

20 The defined types of elements will now be described. These defined type are:

	Allocation	ManagementPolicy
	Assignment	Member
	Context	NamedValue
	DistributionControl	NamedValueList
25	ExecutionConstraints	ProductID
	IntervalTime	Signature

- 69 -

LicenseID	Term
LUDRM	Version
ManagementConstraints	

5 The allocation element records the information concerning a single unit allocation, and is shown in a syntax diagram in Figure 26. Allocation-context specifies the context to which the allocation was made. The allocation-lur field specifies the license unit requirement which applies to the allocation-context; this license unit requirement is calculated without consideration of any allocation sharing which may be possible. The allocation-group-id field identifies the
10 "allocation-group" for the current allocation, in which an unshared allocation will always have an allocation group id of 0; allocations which utilize shared units will have an allocation group id which is shared by all other allocations sharing the same units.

15 The assignment element is shown in syntax diagram in Figure 27. Assigned-units identifies the number of units which are assigned. Assignment-term identifies the start and end of the assignment period. Assignee identifies the context to which the assignment is made.

20 The context element is shown in syntax diagram in Figure 28. The SubContext-type field identifies the type of subcontext, and this type can be either standard or private; if standard, the type value will be taken from the standard-subcontext-type enumeration: (a) network-subcontext means the subcontext value identifies a network; (b) execution-domain-subcontext means the subcontext value is the name of the management domain within which the caller is executing; (d) login-domain-subcontext means the subcontext value is the name of the

management domain within which the user of the caller was originally authenticated or "logged in"; (d) node-subcontext means the subcontext value is the name of a node; (e) process-family-subcontext means the subcontext value is an implementation specific identifier for a group of related processes; (f) process-ID-subcontext means the subcontext value is an implementation specific process identifier; (g) user-name-subcontext means the subcontext value is a user name; (h) product-name-subcontext means the subcontext value is the same as the product name found on the Product Use Authorization; (i) operating-system-subcontext means the subcontext value is a character string representation of the name of the operating system; (j) platform-ID-subcontext means the subcontext value is an identifier that describes the hardware platform supporting the context. The subcontext-value field is the value of the subcontext.

As discussed above, license data is always used or allocated within, or for the benefit of, some named licensing context. This context name is constructed by concatenating the values of all subcontexts into a single context name. A Context Template specifies those components of the context name which should be used in calculating license unit requirements. The management system determines the need to perform a unit allocation each time license units are requested. The full context on whose behalf the allocation should be made is obtained for each requested authorization. The system will mask the full context to exclude all sub-contexts not specified in the context template and then determine if the resulting context already has units allocated to it. If not, units will be allocated according to the specification of the LURDM, otherwise, the units previously allocated will be shared by the new context. Thus, if a given product authorization contains a context specification of NODE + USER_NAME, each context which requests license unit allocations and which has a unique pair

of NODE + USER_NAME subcontext values will require an explicit grant of license units to be made. On the other hand, any contexts which share the same pair of NODE and USER_NAME subcontext values will be able to "share" a single allocation of license units. The requirement for specific allocations of units and the ability to share units is exhibited in Figure 29 which attempts to provide a "snapshot" of the units allocated for the product FOOBAR V4.1 at a particular instance. It is seen from the figure that although presented with five unique full contexts, only four of them are unique when looking only at those portions of each context which are described by the Context Template (ie: NODE + USER_NAME). A unit allocation must be made for each of the four instances of unique contexts, when masked by the Context Template. The fifth context can share allocated units with another context. Thus, the total requirement to support product use as described in this example would be 40-units (ie: four allocations of ten units each). Significant changes in the unit requirements can be achieved by making small modifications to the Context Template. Figure 30 shows the same contexts as in Figure 29 but a Context_Template of NODE. The total unit requirement for this example would be three units (three allocations of ten units each) rather than the forty units required in the previous example.

The distribution control element defines the mechanism that will be used for distributing the subject delegation and records some status information concerning the distribution of that delegation. A syntax diagram of the distribution control element is shown in Figure 31. Distribution-method identifies the means by which the delegation will be distributed, and the alternatives are refresh-distribution, initial=distribution-only, and manual-distribution. Refresh-distribution means the license manager shall be responsible for the initial distribution of the delegation and for ensuring that refresh delegations are

properly distributed. Initial-distribution-only means the license manager shall be responsible for the initial distribution of the delegation, however, distribution of refresh delegations will be made by some other means. Manual-distribution means the distribution of the delegation will be under the control of some other mechanism (perhaps a license asset manager). Current-start-date is the time that the last successful initial or refresh delegation distribution was performed. Current-end-date identifies the last date on which the most recent delegation distribution was performed. Refresh-interval identifies the period of time between attempts to refresh the delegation; the refresh-interval may not be longer than the maximum-delegation-period and should normally be less than that in order to ensure that refresh delegations are distributed prior to the expiration of the previous delegations that they are replacing. Retry-interval identifies the amount of time to wait for an unsuccessful distribution attempt to try again. Maximum-retry-count identifies the maximum number of times that an unsuccessful distribution attempt may be retried. Retries-attempted records the number of unsuccessful retry attempts which have been made since the last successful initial or refresh delegation distribution was performed.

The execution constraints elements place limits on the environments and contexts which may receive allocations. A syntax diagram of the execution constraints element is shown in Figure 32. Operating-system contains a list of zero or more operating systems on which the use of the subject license is authorized; if no operating systems are specified, it is assumed that license use is authorized on all operating systems. Execution-context specifies a list of zero or more full or partial context names which identify the contexts within which products described by the license data may be executed; if no context names are specified, the licensed products may be executed in any context controlled by the licensee.

- 73 -

Environment-list identifies those environments within which the licensed product may be used.

The interval time element is defined by the syntax `IntervalTime ::= UTCTime`.

5 The license ID element uniquely identifies the license data it is associated with, and is described by the syntax diagram of Figure 33. Here issuer uniquely identifies the issuer of the license data as well as the name space within which the LicenseID Number is maintained. While the issuer name will typically be the same as the name of the issuer's company or personal name, this is not a
10 requirement. For instance: The issuer name for Digital Equipment Corporation is "DEC," an abbreviation of the corporate name. Valid contents of the Issuer field are maintained in the an Issuer Registry. The serial-number provides a unique identification or serial number for the license data. The amendment field is an integer which is incremented each time license data is amended by its issuer,
15 with the first version of any license data carries the amendment number 0; an amendment can only be applied to license data if that license data has identical Issuer and Number values and an amendment number less than the number of the amendment to be applied.

20 The license units requirements determination method or LURDM element is shown in syntax diagram in Figure 34. The combination-permitted field indicates whether conforming license managers are permitted to combine together into a common pool the units from different product use authorizations if those produce use authorizations have the same product record value; for example, if combination is permitted and a single license manager discovers in its database

- 74 -

two 500-unit authorizations for the use of DEC Cobol, the license manager would be permitted to combine these two authorizations into a logical grant of 1000 units. The overdraft-limit modifies the behavior of a conforming license management facility in those cases where it is found that there are zero or fewer license units available for use at the time of a request for the allocation or consumption of additional license units. Operation of overdraft is different depending upon whether allocative, or consumptive style is being used. In using with allocative style, an allocation is granted even though the remaining units are zero or less, up to the overdraft-limit. In using with consumptive style, the license is authorized to accumulate a negative balance of license units, up to the overdraft-limit. Overdraft-logging-required indicates whether all license grants which are the result of overdraft use must cause a log record to be generated. When the allocation-size field is non-zero, then all unit allocations and delegations must be made in sizes which are whole number multiples of the allocation-size value. Lurdm-kind identifies the method by which license unit requirements will be calculated once the requirement for an allocation has been discovered, the permitted alternatives being (a) LURT which specifies that license unit requirements are to be determined by lookup in the LURT which is associated with the current license, (b) Constant which specifies that license unit requirements are constant for all platforms on which the licensed product or product feature may run, and (c) Private-LURDM which specifies that license unit requirements are to be determined by the licensed product, not by the license management facility. The named-lurt-id specifies the name of the LURT table to be used in determining license unit requirements if the LURDM-kind is specified as LURT; if the LURDM-kind is specified as LURT and no table is explicitly named, the name of the table to be used is constructed from the issuer name on the product use authorization. Lurdm-value specifies the LURT column to be

-75-

used when LURDM-kind = LURT; however, when LURDM-kind = Constant, the Lurdm-value field contains the precise number of units to be allocated or consumed. Default-unit-requirement specifies the unit requirement value to be used when the appropriate LURT does not have a row corresponding to the appropriate platform ID; when specified on a product use authorization with Style = Allocative, the context template will change to Process + Product_Specific and the Duration will change to Transaction in cases of unrecognized Platform ID's.

The management constraints element is shown in a syntax diagram in Figure 35. The management-context field specifies a list of zero or more partial context names which identify the specific contexts within which the license data may be managed. If no management contexts are specified, the license data may be managed within any context controlled by the licensee. The contexts used in specifying Management Context Constraints may only contain the Network, Domain, and Node subcontexts. Specifying a list of management contexts does not effect whether or not the license data can be used within other contexts. For example, unless otherwise restricted, license data with a specified management context can be remotely accessed from or delegated to other nodes in a network. The management-scope field defines the maximum permitted size of the license management domain within which the license data may be managed or distributed, these being single-platform, management-domain, or entire-network. Single-platform constrains the license management domain for the subject license data to be no larger than a single platform. Management-domain constrains the license management domain for the subject license data to be no larger than a single management domain. Entire-network constrains the license management domain for the subject license data to be no larger than a single wide area network; that

network which contains the platform on which the license units were initially loaded. Although technology may not exist to detect the interorganizational boundaries of a wide area network (i.e., what is on the Internet as opposed to being on a company's own network), the assumption is that interorganization and internetwork sharing of licenses will normally be considered a violation of license terms and conditions. The backup-permitted field indicates if the Issuer has authorized the use of backup delegations for this data. Delegation-permitted indicates if the Issuer has authorized the licensee to delegate this data. Maximum-delegation-period identifies the longest interval during which a delegation may be valid; by default, delegations have a life of 72-hours.

The major elements of the management policy specification are shown in Figure 3, as previously discussed. A syntax diagram for the management policy element is shown in Figure 36. For the Style field, three fundamental styles of license management policy are supported, allocative, consumptive, and private-style, as explained above. Only one of these styles may be assigned to any single product use authorization. The Context-template specifies those components (sub-contexts) of the execution-context name which should be used in determining if unit allocations are required. The Duration defines the duration of an allocation of license units to a specific context or the duration of the period which defines a valid consumptive use. For durations of type "Assignment," the specification of a Reassignment Constraint is also provided for. Three types of Duration_Kind are supported, these being Transaction, Assignment and Immediate, as explained above. The lur-determination-method stores information used in calculating the number of units that should be allocated or consumed in response to a license request. The allocation-sharing-limit identifies the largest number of execution contexts that may share an allocation made under this management policy; an

allocation-sharing-limit of 0 indicates that the number of execution contexts that may share an allocation is unlimited. The reassignment-constraint specifies a minimum duration of assignment; although there is normally no constraint placed on how frequently granted units may be reassigned, an issuer may constrain
5 reassignment by specifying this minimum duration of an assignment, in which case reassignment of assigned units will not be supported until the amount of time specified in the Reassignment Constraint has passed. If an assignment of some particular set of units has been delegated and the delegation period for that delegation has not terminated, cancellation of the delegation must be performed
10 prior to reassignment.

The member element identifies a specific licensed product which may be part of a calling authorization or group definition, and is shown in syntax diagram in Figure 37. Member-product identifies the product which is a member. Member-signature is constructed from the product and token fields of the called
15 member structure as well as the product and issuer fields of the calling product. Member-token provides the data which should be used as the product token for this member.

Named values are data elements with a character string tag that identifies the data element, and have a syntax as shown in Figure 38, which also shows the
20 syntax for ValueData and named value list. A named value list models a list of named values, with an example being shown in Figure 39. In Figure 38, Value-Name uniquely identifies the value; no standard value names are defined, and the period character can be used as a part of the value name to form a hierarchical tag registry at the discretion of the issuer. Value-data is the data that has been
25 named; data types are selected from the possible Value Data types, seen in the

Figure. Value-boolean means the named data is a boolean value. Value-integer means the named data is an integer value. Value-text means the named data is a StringList value. Value-general means the named data is a stream of bytes in any format. Value-list means the named data is a list of named data values.

5 The product ID explicitly identifies the product which is the subject of the license data with which it is associated, with the syntax for ProductID being shown in Figure 40. The version and release date fields provide a mechanism for defining which specific instances of the licensed product are described in the associated license data. The Producer field is a registered name which identifies
10 the producer of the licensed feature; in the case of Group Names, the Producer is always also the Issuer of the group. The Product-name identifies a licensed software feature. The First-version identifies the earliest version of the product whose use is authorized. The Last-version identifies the latest version of the product whose use is authorized. The First-release-date identifies the earliest
15 release of the product whose use is authorized. The Last-release-date identifies the latest release of the product whose use is authorized. Conforming license managers are required to interpret the contents of these fields in the most restrictive way possible. Thus, if a license is issued with Last-version = 3.0 and a Last-release-Date of 1-Jan-1991, then the use of version 2.0 of the licensed
20 product would be unauthorized if it had a release date of 2-Jan-1991. If either a First-version or First-release-date is specified without a matching Last-version or Last-release-date, use of the produce is authorized for all versions or release dates following that specified. Similarly, if either a last-version or Last-release-date is specified without a matching First-version or First-release-date, use of the produce
25 is assumed to be authorized for all versions or release dates prior to that specified. Issuers should typically only specify one of either First-version or First-release-

date. This is the case since it is anticipated that these fields will typically refer to events which occurred prior to the moment of license data issuance. Thus, it should normally be possible for the issuer to state unambiguously with only one of these two fields which is the oldest implementation of the product that is to be authorized. The architecture does permit, however, both fields to be used in a single product authorization.

The signature element is used to establish the integrity and authorship of the license data with which it is associated. A syntax diagram for the signature element is shown in Figure 41. The Signature-algorithm field identifies the registered algorithm that was used to produce the digital signature. Signature-parameters are the values of the algorithm's parameters that are to be used; the need for and syntax of parameters is determined by each individual algorithm. Signature-value is an enciphered summary of the information to which the signature is appended; the summary is produced by means of a one-way hash function, while the enciphering is carried out using the secret key of the signer (Issuer).

The term element defines an interval during which the license data is valid, and is shown in syntax diagram form in Figure 42. The fields are start-date and end-date. Start-date identifies the first date of the term; if not specified, the license data is considered valid on any date prior to the end-date. End-date identifies the last date of the term; if not specified, the license data is considered valid on any date after the Start-date. While the Start-date is always either omitted or specified as an absolute date, the End-date can be either absolute or relative. If the End-date is specified as a relative or "interval" date and the Start-date has been omitted, the date of license registration will be used as the effective

- 80 -

5 start date in computing the valid term of the license data. It should be noted that the system does not specify the mechanism by which system dates are maintained by platforms supporting system components. Instead, the system always accepts that system time returned to it as correct. Thus, the reliability of the management of license data which specifies terms is dependent on the time management function of the underlying platform.

10 The version element identifies a four-part version of the licensed software product or feature. A syntax diagram of the version element is shown in Figure 43. The schematics of each of the four parts is not detailed, but it is required that producers who wish to permit version ranges to be specified on product use authorizations ensure that the collating significance of the four parts is maintained. When comparing versions, Part-1 is considered first, then Part-2, then Part-3, and finally, Part-4. Part-1 identifies a major modification to the versioned object. Part-2 identifies a modification to the versioned object which is less significant than a modification which would cause a change in the Part-1 value. Part-3 identifies a modification to the versioned object which is less significant than a modification which would cause a change in the Part-2 value. Part-4 identifies a modification to the versioned object which is less significant than a modification which would cause a change in the Part-3 value.

20 FILTERS

An important feature is the use of filters in the license management program 11, including the client interface 31 and the management interface 33. A filter is used to select items in the license database 23, for example. Various

- 81 -

selection mechanisms are used in picking out or doing lookups in database technology; filters are one of them. The filter engine used in the license management system 11 of Figure 1 is generally of a known construction, with the exception of the select filter item type as will be described, which allows a complex rather than a flat data format to be selected from. The feature that is of importance to this embodiment is the way of specifying items as an input to the filter function , rather than the filter function itself. Thus, there is described below a template for specifying input to the filter engine. This is as if a form were used as the input, with blanks on the form; by filling in certain blanks these would be the items selected on, the blanks not filled in would be "don't care".

An instance of the class *filter* is a basis for selecting or rejecting an object on the basis of information in that object. At any point in time, a filter has a value relative to every object - this value is false, true or undefined. The object is selected if and only if the filter's value is true. This concrete class has the attributes of its superclass - *Object* - and the specific attributes listed in the table of Figure 44.

A filter is a collection of simpler filters and elementary filter-items together with a Boolean operation. The filter value is undefined if and only if all the component filters and filter-items are undefined. Otherwise, the filter has a Boolean value with respect to any object, which can be determined by evaluating each of the nested components and combining their values using Boolean operation (components whose value is undefined or ignored). The attributes specific to *filter* as shown in Figure 44 are (a) *filter items* which are a collection of assertions, each relating to just one attribute of an object, (b) *filters* which are a

collection of simple filters, and (c) *filter type* which is the filter's type, of one of the following values: And, Or, Not.

5 An instance of the class *filter item* is a component of a *filter*. It is an assertion about the existence or values of a single attribute of a license data object or one or its subobjects. This concrete class has the attributes of its superclass - *object* - and the specific attributes listed in the table of Figure 45.

10 The value of a filter item is undefined if: (a) the Attribute Types are unknown, or (b) the syntax of the Match Value does not conform to the attribute syntax defined for the attribute type, or (c) a required Attribute is not provided. The attributes specific to *filter item* as shown in Figure 45 are (a) *filter item type* which identifies the type of filter item and thereby the nature of the filter, and its value must be one of

	equality	less
	inequality	present
15	greater or equal	select
	less or equal	request candidates
	greater	simulate request

20 (b) *attribute type* which identifies the type of that attribute whose value or presence is to be tested; the value of All Attributes may be specified, (c) *match value* which is the value which is to be matched against the value of the attribute, (d) *filter* which identifies the filter to be used in evaluating a selected subobject of the current object; the filter is ignored if the *filter item type* is not *select* or if the specified attribute type is not present in the object, and upon evaluation of the *filter* the value of *filter item* will be set to that of the *filter*, (e) *initial substring*, if
25 present, this is the substring to compare against the initial portion of the value of

- 93 -

the specified attribute type, (f) *substring*, if present, this is the substring(s) to compare against all substrings of the value of the specified attribute type, (g) *final substring*, if present, this is the substring to compare against the final portion of the value of the specified attribute type, and (h) *license request*, if present, this is license request against which the appropriate license data objects should be evaluated; this attribute may only be specified if the value of the filter item type is either Request Candidates or Simulate Request.

An instance of enumeration syntax *Filter Type* identifies the type of a filter. Its value is chosen from one of the following: (a) *And* means the filter is the logical conjunction of its components; the filter is true unless any of the nested filters or filter items is false, or if there are no nested components, the filter is true; (b) *Or* means the filter is the logical disjunction of its components; the filter is false unless any of the nested filters or filter items is true, or, if there are no nested components, the filter is false; (c) *Not* means the result of the filter is reversed; there must be exactly one nested filter or filter item, and the filter is true if the enclosed filter or filter item is false, and is false if the enclosed filter or filter item is true.

An instance of enumeration syntax *Filter Item Type* identifies the type of a filter item. Its value is chosen from one of the following: (a) *Equality* which means the filter item is true if the object contains at least one attribute of the specified type whose value is equal to that specified by Match Value (according to the equality matching rule in force), and false otherwise; (b) *Inequality* which means the filter item is true if the object contains at least one attribute of the specified type whose value is not equal to that specified by Match Value (according to the equality matching rule in force), and false otherwise; (c) *Greater*

5 *or Equal* which means the filter item is true if the object contains at least one attribute of the specified type whose value is equal to or greater than the value specified by Match Value (according to the matching rule in force), and false otherwise; (d) *Less or Equal* which means the filter item is true if the object contains at least one attribute of the specified type whose value is equal or less than the value specified by Match Value (according to the matching rule in force), and false otherwise; (e) *Greater* which means the filter item is true if the object contains at least one attribute of the specified type whose value is greater than the value specified by Match Value (according to the matching rule in force), and false otherwise; (f) *Less* which means the filter is true if the object contains at least one attribute of the specified type, whose value is less than the value specified by Match Value (according to the matching rule in force), and false otherwise; (g) *Present* which means the filter item is true if the object contains at least one attribute of the specified type, and false otherwise; (h) *Select* which means the filter item is true if the object contains at least one attribute of the specified type which has an object syntax and when the Filter is evaluated against the attributes of that object the Filter is true, and false otherwise; (i) *Request Candidates* which means the filter item is true if the object against which it is evaluated is one which could be used to provide some or all of the units requested by the specified License Request; the evaluation is made independently of any outstanding allocations or preallocations; and (j) *Simulate Request* which means the filter item is true if the object against which it is evaluated is one which would be used to provide some or all of the units requested by the specified License Request.

25 The Request Candidates and Simulate Request filter item types are of special use in testing and prototyping of systems by a license manager at a

licensee's site. For example, the license manager can simulate the effect of potential assignments, the effect of a population of certain types on a network, etc.

As an example, Figure 46 shows how a filter may be constructed to identify "All Product Use Authorizations issued by Digital for the Product 'Amazing Graphics System' which contains a calling authorization for Digital's 'Amazing Database' Product". This example is in the international standard format referred to as X.409 as mentioned above.

Filters can also be used in a request allocation, being specified in a request extension as explained above. That is, a filter is one of the optional items in a request extension. For example, if a user wanted to use a version of WordPerfect with French language extension, and there were version with and without on the network, his request allocation would have a request extension that specified a filter for "French" in the token field. In this manner, a product can describe itself more richly. The filter in the request extension can be a Required filter or a Preferred filter, meaning the feature such as "French" is either absolutely necessary, or merely the preferred.

While this invention has been described with reference to specific embodiments, this description is not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as other embodiments of the invention, will be apparent to persons skilled in the art upon reference to this description. It is therefore contemplated that the appended claims will cover any such modifications or embodiments as fall within the true scope of the invention.

- 86 -

WHAT IS CLAIMED IS:

1 1. A method of managing use of licensed software items, said
2 software items separately executable on a computer system or
3 accessible by said computer system, the computer system including
4 a processor and one or more nodes, comprising the steps of:

5 maintaining by said processor a store of license
6 authorizations for said software items; each license authorization
7 including an indication of license management policy for a software
8 item, said indication having a plurality of sets of policy
9 components, said sets of policy components granting alternatives of
10 specified restrictive rights to selectively access and execute said
11 software items in said system; said indication of license
12 management policy being in the format of an encoded document of a
13 data type consisting of an ordered sequence of elements;

14 accessing said store by said processor to modify in said store
15 one or more of said specified restrictive rights of said policy
16 components of an identified license authorization;

17 accessing said store by said processor using a filter to
18 obtain information from said license authorization for a selected
19 software item, in response to a request from a node, and

20 comparing an identification of said node and said software
21 item with said information, to produce and send to said node a
22 grant or refusal of said request.

1 2. A method according to claim 1 including the step of
2 receiving said license authorizations , for storing in said store,

1 from a license grantor external to said processor, and wherein said
2 step of accessing said store to modify in said store one or more of
3 said specified restrictive rights employs management functions
4 executable on said processor but not on said nodes or said license
5 grantor to identify a license authorization in said store.

1 3. A method according to claim 1 wherein said indication is
2 in the format of an encoded document of a data type consisting of
3 an ordered sequence of three elements, the three elements including
4 a document descriptor, a document header and the document content.

1 4. A method according to claim 1 wherein said filter
2 specifies one or more of said attributes and a Boolean operator for
3
4 each selected attribute.

1 5. A method according to claim 2 wherein said step of
2 accessing said store to modify one or more of said policy
3 components is to allow grant of rights to use which are more
4 restrictive than said specified restrictive rights.

1 6. A method according to claim 2 including the steps of:

2

3 sending a request by a user of one of said software items to
4 obtain permission to use said software item; said request
5 identifying the user and said software item;

1 accessing said store to obtain information from said license
2 authorization for said software item, in response to said request,
3 and comparing said identification of said user and said software
4 item with said information, to produce a grant or refusal of said
5 request for sending to said user.

1 7. A method according to claim 6 wherein said store is
2 maintained by a license server, and said request is sent to said
3 server and wherein said request is in the form of a remote
4 procedure call, and said grant or refusal sent to said user is a
5 return of said procedure call.

1 8. A method according to claim 7 wherein said license
2 authorization is a data arrangement specified as a product use
3 authorization, and said product use authorization is received by
4 said server from an issuer, and wherein said server and said users
5 are nodes on a computer network.

1 9. A method according to claim 2 wherein said policy
2 components include a termination date, and said management
3 functions can modify said termination date to an earlier
4 termination date and wherein said policy components include a right
5 of delegation of a right to grant said requests to another server,
6 and said management functions can modify said right of delegation
7 to remove said right of delegation.

1 10. A method according to claim 2 including storing in
2 association with said license authorization a number of management
3 attributes, and said management functions being able to modify said
4 management attributes.

1 11. A method according to claim 10 wherein said management
2 attributes include a reservation of units of license use granted by
3 said license authorization so that said units will not be granted
4 to a user in response to said request, and wherein said management
5 attributes include an allocation of units of license use to a
6 specific context.

1 12. A method according to claim 10 wherein said management
2 attributes include an allocation period which is the minimum
3 duration of an allocation of units, and wherein said management
4 attributes include permission to enable a backup delegation of the
5 right to grant said requests.

1 13. A system for managing use of licensed software products,
2 comprising: means for maintaining a store of license documents, one
3 for each said product; each license document including an
4 indication of license policy having plurality of sets of policy
5 components granting specified restrictive rights to use said
6 software products, said policy components in each set providing
7 alternatives;

8 a management interface for accessing said store to modify

1 selected ones of said components of an identified license
2 authorization.

1 14. A system according to claim 13 including:

2 means for sending a request from a user of one of said
3 products to obtain permission to use said product; said request
4 identifying the user and said product;

5 means for accessing said store to obtain information from said
6 license document for said product, in response to said request, and
7 for comparing said identification of said user and said product
8 with said information, and with constraints imposed by said policy
9 components, to produce a grant or refusal of said request and send
10 said grant or refusal to said user.

1 15. A system according to claim 13 wherein said management
2 interface can modify said selected ones of said components to allow
3 grant of rights to use which are more restrictive than said
4 specified restrictive rights and wherein said means for
5 maintaining, and said means for accessing and sending to said user
6 are all located at a server on a distributed network, and said
7 means for sending a request is located at a user node on said
8 network.

1 16. A system according to claim 14 wherein said request is in
2 the form of a remote procedure call, and said grant or refusal sent
3 to said user is a return of said procedure call, and wherein said

1 license document is a data arrangement specified as a product use
2 authorization, and said product use authorization is received by
3 said server from a license issuer.

1 17. A system according to claim 13 wherein said policy
2 components include a termination date, and said management
3 functions can modify said termination date to an earlier
4 termination date, and wherein said policy components include a
5 right of delegation of a right to grant said requests to another
6 server, and said management functions can modify said right of
7 delegation to remove said right of delegation.

1 18. A system according to claim 15 including means for storing
2 in association with said license authorization a number of
3 management attributes, wherein said management functions are able
4 to modify said management attributes and wherein said management
5 attributes include a reservation of units of license use granted by
6 said license authorization so that said units will not be granted
7 to a user in response to said request.

1 19. A system according to claim 18 wherein said management
2 attributes include an allocation of units of license use to a
3 specific context.

1 20. A system according to claim 18 wherein said management
2 attributes include an allocation period which is the minimum

1 duration of an allocation of units, and include permission to
2 enable a backup delegation of the right to grant said requests.

1 21. A method according to claim 3 wherein said document
2 descriptor includes an encoding method version number, and encoder-
3 identifier and an encoder-name, and wherein said document-header
4 includes a title, an author, a version and a date for the software
5 item.

1 22. A method according to claim 3 wherein said document
2 content includes at least one of the following:

- 3 a product-use-authorization;
- 4 a license-use-requirements-table;
- 5 a group-definition;
- 6 a key-registration;
- 7 a delegation.

1 23. A method according to claim 3 wherein said document-
2 content includes a license-data-header, and said license-data-
3 header describes the parties to the license document, the term of
4 the agreement and constraints that may have been placed on
5 management of the license data.

1 24. A method according to claim 3 wherein said document-
2 content includes management-info, where the management-info may
3 include at least one of the following:

1 an assignment;
2 a reservation;
3 a delegation;
4 a backup delegation;
5 an allocation;
6 a registration date;
7 a registrar;
8 a comment;
9 a termination-date.

1 25. A method according to claim 3 wherein:
2 said document descriptor includes an encoding method
3 version and a date for the software item;
4 said document content may include at least one of the
5 following: a product-use-authorization, a license-use-requirements-
6 table, a group-defination, a key-registration, and a delegation;
7 said document-content selectively includes a license-
8 data-header, and said license-data-header describes the parties to
9 the license document, the term of the agreement and constraints
10 that may have been placed on management of the license data;
11 said document-content may have been placed on management
12 of the license data;
13 said document-content selectively includes management-
14 info, where the management-info may include at least one of the
15 following: an assignment, a reservation, a delegation, a backup
16 delegation, an allocation, a registration date, a registrar, and a

1 comment.

1 26. A method according to claim 3 wherein said store is
2 maintained by a license server, and said request is sent to said
3 server, and wherein said server and said users are nodes on a
4 computer network.

1 27. A method according to claim 3 wherein said request is in
2 the form of a remote procedure call, and said grant or refusal sent
3 to said user is a return of said procedure call, and wherein said
4 license authorization is received by said server from an issuer.

1 28. A method according to claim 3 including the steps of:
2 sending a request by a user of one of said software items to obtain
3 permission to use said software item; said request identifying the
4 user and said software item;
5 sending said grant or refusal to said user.

1 29. Apparatus for managing use of licensed software items,
2 comprising:
3 means for maintaining a store of license authorizations
4 for said software items; each license authorization including an
5 indication of license management policy for a software item, said
6 indication being in the format of an encoded document of a data
7 type consisting of an ordered sequence of three elements, the three
8 elements including a document descriptor, a document header and the

1 document content;

2 means for sending a request by a user of one of said
3 software items to obtain permission to use said software item; said
4 request identifying the user and said software item;

5 means for accessing said store to obtain information from
6 said license authorization for said software item, in response to
7 said request, and comparing said identification of said user and
8 said software item with said information, to produce a grant or
9 refusal of said request;

10 means for sending said grant or refusal to said user.

1 30. Apparatus according to claim 29 wherein said document
2 descriptor includes an encoding method version number, and an
3 encoder-identifier and an encoder-name, and wherein said document-
4 header includes a title, an author, a version and a date for the
5 software item.

1 31. Apparatus according to claim 29 wherein said document
2 content includes at least one of the following:

3
4 a product-use-authorization;
5 a license-use-requirements-table;
6 a group-definition;
7 a key-registration;
8 a delegation.
9

1 32. Apparatus according to claim 29 wherein said document-
2 content includes a license-data-header, and said license-data-
3 header describes the parties to the license document, the term of
4 the agreement and constraints that may have been placed on
5 management of the license data.

1 33. Apparatus according to claim 29 wherein said document-
2 content includes management-info, where the management-info may
3 include at least one of the following:
4 an assignment;
5 a reservation;
6 a delegation;
7 a backup delegation;
8 an allocation;
9 a registration date;
10 a registrar;
11 a comment;
12 a termination-date.

1 34. Apparatus according to claim 29 wherein:
2 said document descriptor includes an encoding method
3 version number, and encoder-identifier and an encoder-name;
4 said document-header includes a title, an author, a
5 version and a date for the software item;
 said document content may include at least one of the
following: a product-use-authorization, a license-use-requirements-

table, a group-definition, a key-registration, and a delegation;

said document-content may include a license-data-header, and said license-data-header describes the parties to the license document, the term of the agreement and constraints that may have been placed on management of the license data;

said document-content may include management-info, where the management-info may include at least one of the following: an assignment, a reservation, a delegation, a backup delegation, an allocation, a registration date, a registrar, and a comment.

1

2

35. Apparatus according to claim 29 wherein said store is maintained by a license server, and said request is sent to said server, and wherein said request is in the form of a remote procedure call, and said grant or refusal sent to said user is a return of said procedure call.

3

4

5

6

1

36. Apparatus according to claim 29 wherein said license authorization is received by said server from an issuer, and wherein said server and said users are nodes on a computer network.

2

3

1

37. A method of storing license documents by a server for a license management system, comprising the steps of:

2

3

maintaining a store of license documents for software items; each license document including an indication of license management policy for a software item, said indication being in the format of an encoded document of a data type consisting of an ordered

4

5

6

1 sequence of three elements, the three elements including a document
2 descriptor, a document header and the document content;

3 accessing said store to obtain information from a selected one
4 of said license documents for a software item, in response to a
5 request, and referencing said indication of license management
6 policy, to produce a grant or refusal of said request.

1 38. A method according to claim 37 wherein said document
2 descriptor includes an encoding method version number, an encoder-
3 identifier and an encoder-name, and wherein said document-header
4 includes a title, an author, a version and a date for the software
5 item.

1 39. A method according to claim 37 wherein said document
2 content includes at least one of the following:

- 3 a product-use-authorization;
4 a license-use-requirements-table;
5 a group-definition;
6 a key-registration;
7 a delegation.

1 40. A method according to claim 4 wherein said step of
2 selecting by a filter may select on one or more of the attributes:
3 issuer, producer, product name, product use authorization, calling
4 authorization, and wherein said store is maintained by a license
5 server, and said request is sent to said server.

1 41. A method according to claim 4 wherein said request is in
2 the form of a remote procedure call, and said grant or refusal sent

1 to said user is a return of said procedure call.

1 42. A method according to claim 40 wherein said license
2 authorization is a data arrangement specified as a product use
3 authorization, and said product use authorization is received by
4 said server from an issuer, and wherein said server and said users
5 are nodes on a computer network.

1 43. Apparatus for managing use of licensed software items,
2 comprising:

3 means for maintaining a store of license authorizations for
4 said software items; each license authorization including an
5 indication of license management policy for a software item, said
6 indication being an encoded document containing a number of
7 attributes defining said license policy;

8 filter means for selecting from said store, said filter means
9 specifying one or more of said attributes and a Boolean operator
10 for each selected attribute;

11 means for sending a request by a user of one of said software
12 items to obtain permission to use said software item; said request
13 identifying the user and said software item;

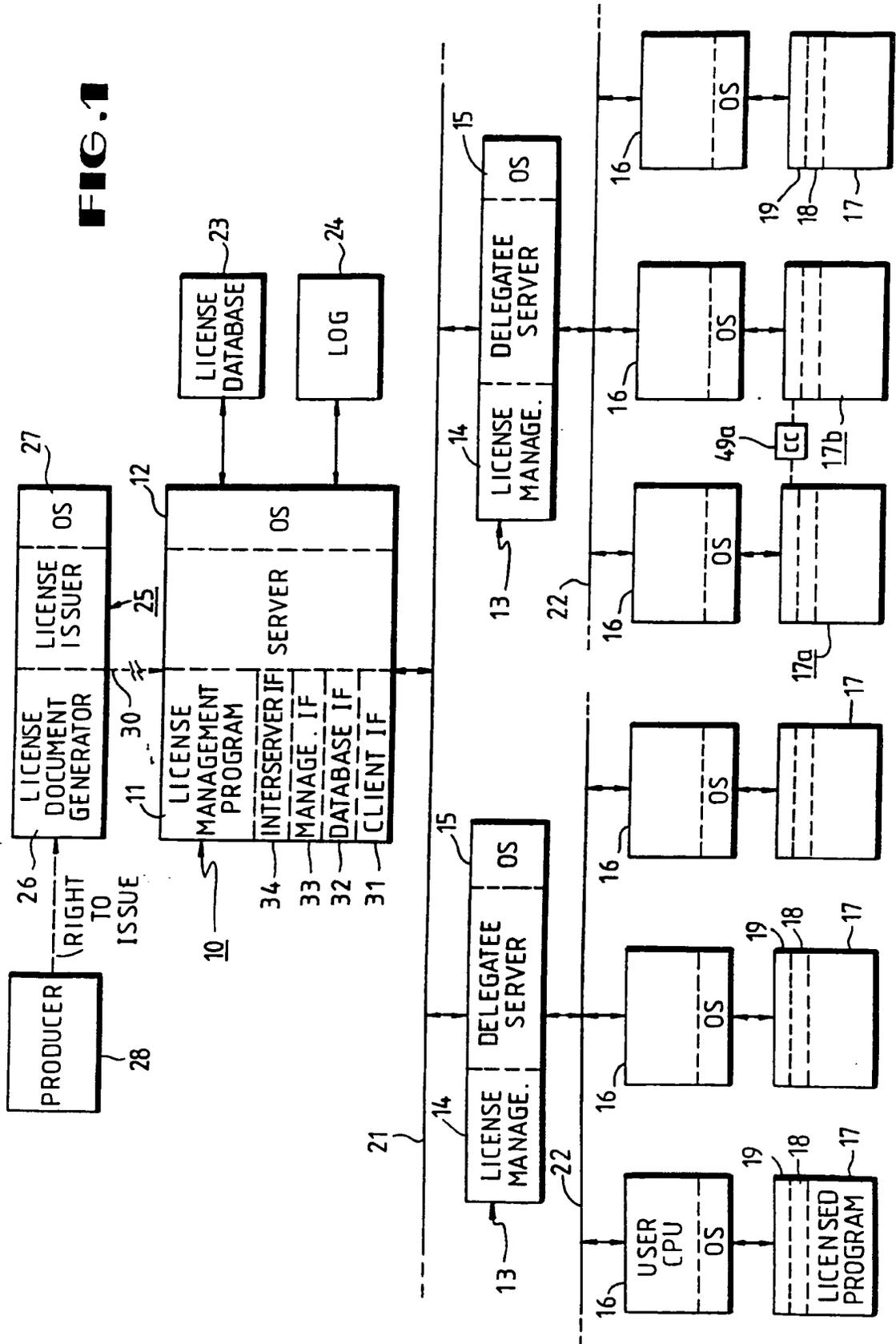
14 means for accessing said store to obtain information from said
15 license authorization for said software item, in response to said
16 request, and comparing said identification of said user and said
17 software item with said information, to produce a grant or refusal
18 of said request; and

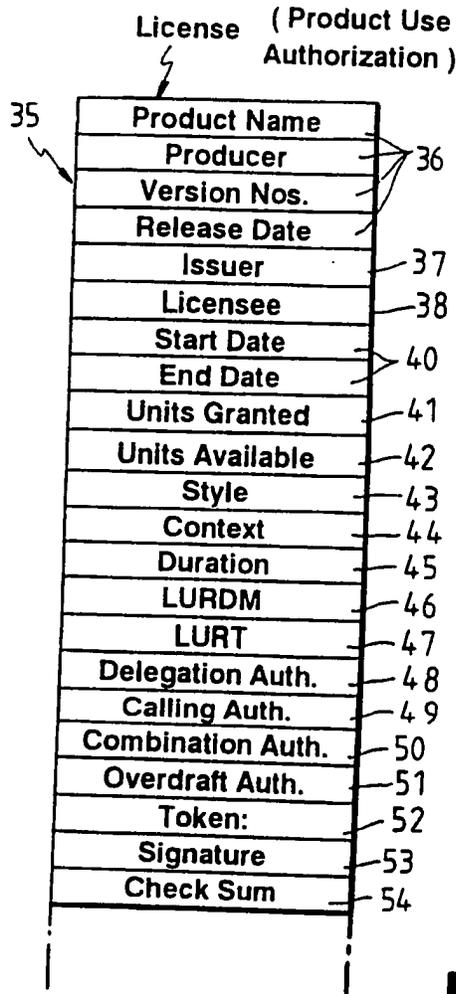
1 means for sending said grant or refusal to said user.

1 44. Apparatus according to claim 43 wherein said filter means
2 may select on one or more of the attributes: issuer, producer,
3 product name, product use authorization, calling authorization, and
4 wherein said store is maintained by a license server, and said
5 request is sent to said server, and wherein said request is in the
6 form of a remote procedure call, and said grant or refusal sent to
7 said user is a return of said procedure call.

1 45. Apparatus according to claim 43 wherein said license
2 authorization is a data arrangement specified as a product use
3 authorization, and said product use authorization is received by
4 said server from an issuer, wherein said server and said users are
5 nodes on a computer network.

FIG. 1





License Unit Requirements Table			
Row Selector	Columns		
Platform ID	A	B	C
PC-0	10	230	-1
PC-1	12	230	-1
VAX 6210	158	300	150

FIG. 4

FIG. 2

43 Style	44 Context	45 Duration	46 LURDM
Allocative	Network	Transaction	Constant
Consumptive	Execution_Domain	Assignment	Table Lookup
Private	Login_Domain	Immediate	Private
	Node_ID		
	Process_Family		
	Process		
	User_Name		
	Product_Name		
	Operating_System		
	Platform_ID		
	Private		

FIG. 3

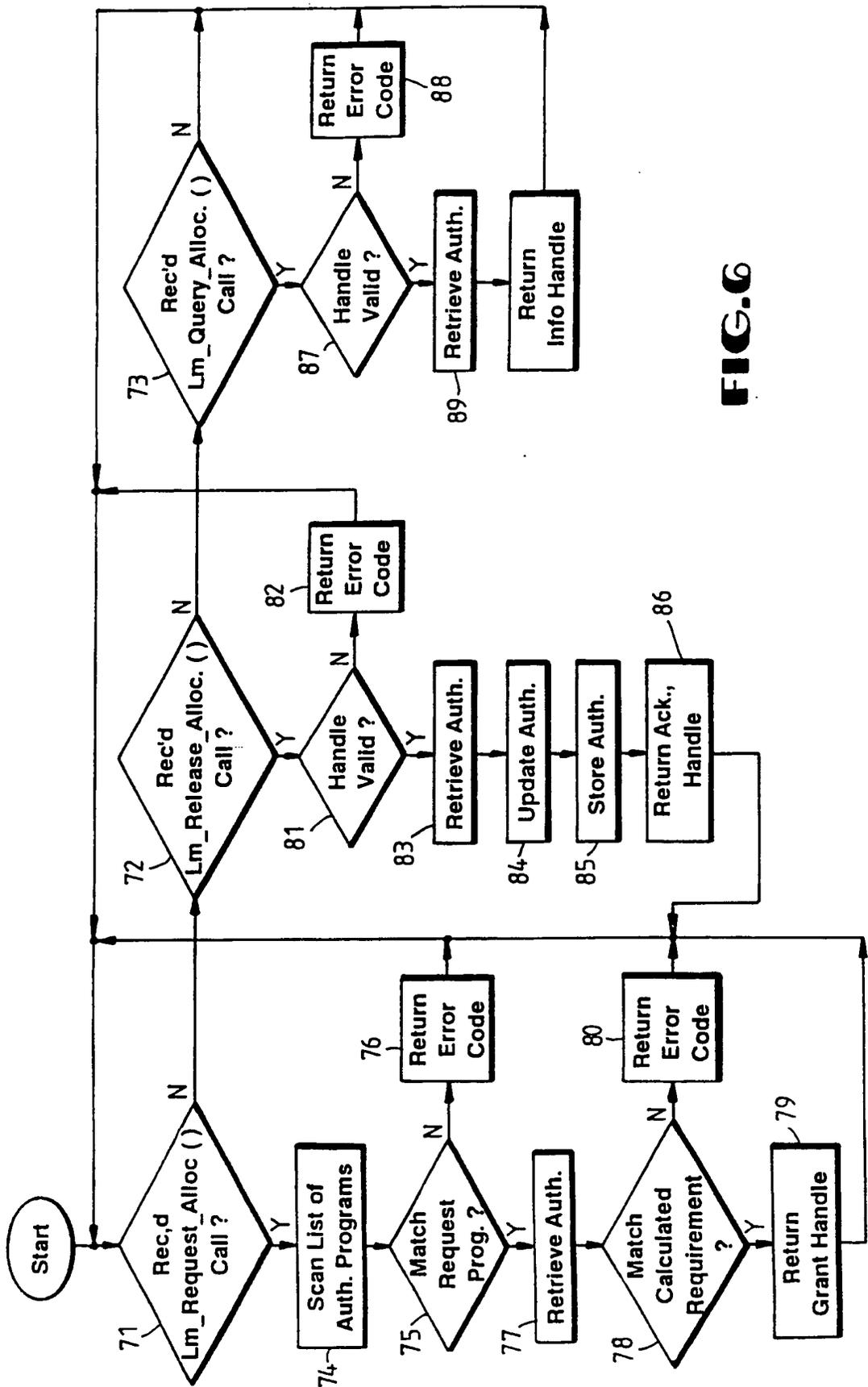


FIG. 6

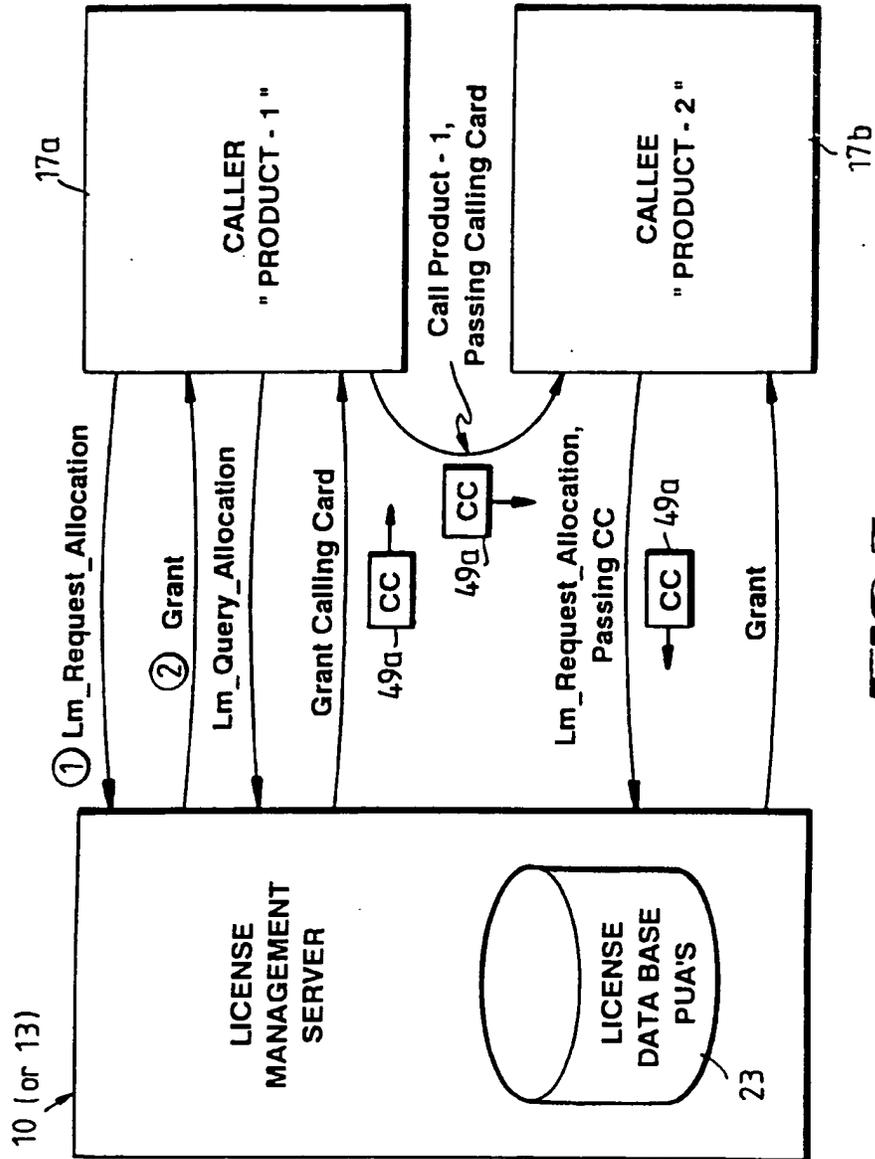


FIG.7

SUBSTITUTE SHEET

```

Object Identifier Value ::= {
    iso(1)
    identified-organization(3)
    icd-ecma(12)
    member-company(2)
    dec(1011)
    data-syntaxes(1)
    cda(3)
    ldif(17)
}

Object Identifier Encoding ::= {
    0x6, 0x8, 0x2B, 0xC, 0x2,
    0x87, 0x73, 0x1, 0x3, 0x11
}

```

FIG. 8 LDIF Object Identifier

```

LDIFDocument ::= [PRIVATE 16373] IMPLICIT SEQUENCE {
  document-descriptor [0] IMPLICIT DocumentDescriptor OPTIONAL,
  document-header [1] IMPLICIT DocumentHeader OPTIONAL,
  document-content [2] IMPLICIT DocumentContent
}

```

FIG. 9 LDIF Document Syntax Diagram

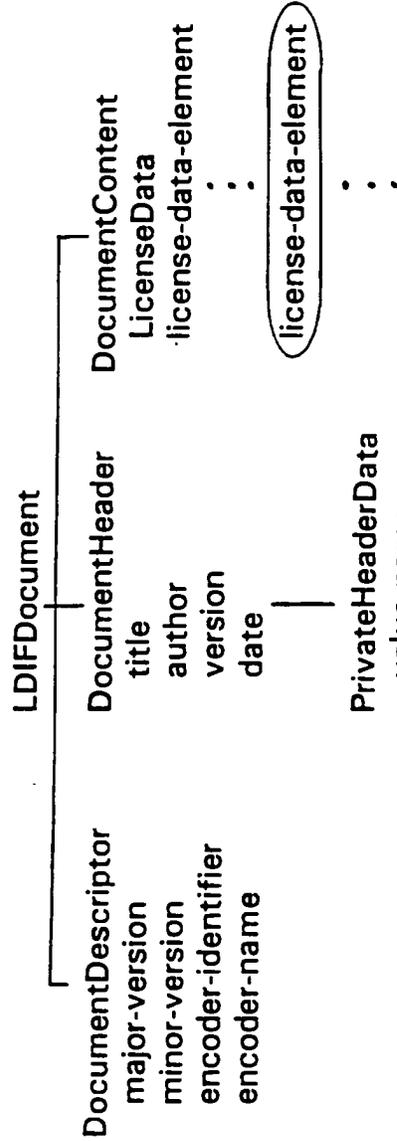


FIG. 10 LDIF Document Structure

```

DocumentDescriptor ::= SEQUENCE {
    major-version [0] IMPLICIT INTEGER OPTIONAL,
    minor-version [1] IMPLICIT INTEGER OPTIONAL,
    encoder-identifier [2] IMPLICIT Character-String OPTIONAL,
    encoder-name [3] IMPLICIT Character-String OPTIONAL
}

```

FIG. 11 Document Descriptor Syntax Diagram

```

Pakgen DocumentDescriptor ::= {
    major-version 1,
    minor-version 0,
    encoder-identifier "PAKGEN",
    encoder-name {Character-String "PAK Generator V1.0"}
}

```

FIG. 12 Document Descriptor Example

SUBSTITUTE SHEET

```

DocumentHeader ::= SEQUENCE {
  private-header-data
  title
  author
  version
  date
}
SEQUENCE {
  [0] IMPLICIT NamedValueList OPTIONAL,
  [1] IMPLICIT Character-String OPTIONAL,
  [2] IMPLICIT Character-String OPTIONAL,
  [3] IMPLICIT Character-String OPTIONAL,
  [4] IMPLICIT UTCTime OPTIONAL
}
    
```

FIG. 13 Document Header Syntax Diagram

```

example-header document-header ::= {
  title {Character-String "PAKGEN Licenses with Associated LURT data"}
  author {Character-String "Tom Jones, Foobar, Inc. License Department"}
  version {Character-String "VO.1"}
  date "198801021100-0500"
}
    
```

FIG. 14 Document Header Example

```

Document Content      ::= SEQUENCE OF LicenseData

LicenseData
  license-data-header
  license-body
  product-use-authorization
  license-units-requirements-table
  group-definition
  key-registration
  issuer-delegation
  license-delegation
  backup-delegation
  management-info
}
  ::= SEQUENCE {
    [0] IMPLICIT LicenseDataHeader,
    [1] CHOICE {
      [0] IMPLICIT ProductUseAuthorization,
      [1] IMPLICIT LURT,
      [2] IMPLICIT GroupDefinition,
      [3] IMPLICIT KeyRegistration,
      [4] IMPLICIT IssuerDelegation,
      [5] IMPLICIT LicenseDelegation,
      [6] IMPLICIT BackupDelegation
    },
    [2] IMPLICIT ManagementInfo OPTIONAL
  }

```

FIG. 15 Document Content Syntax Diagram

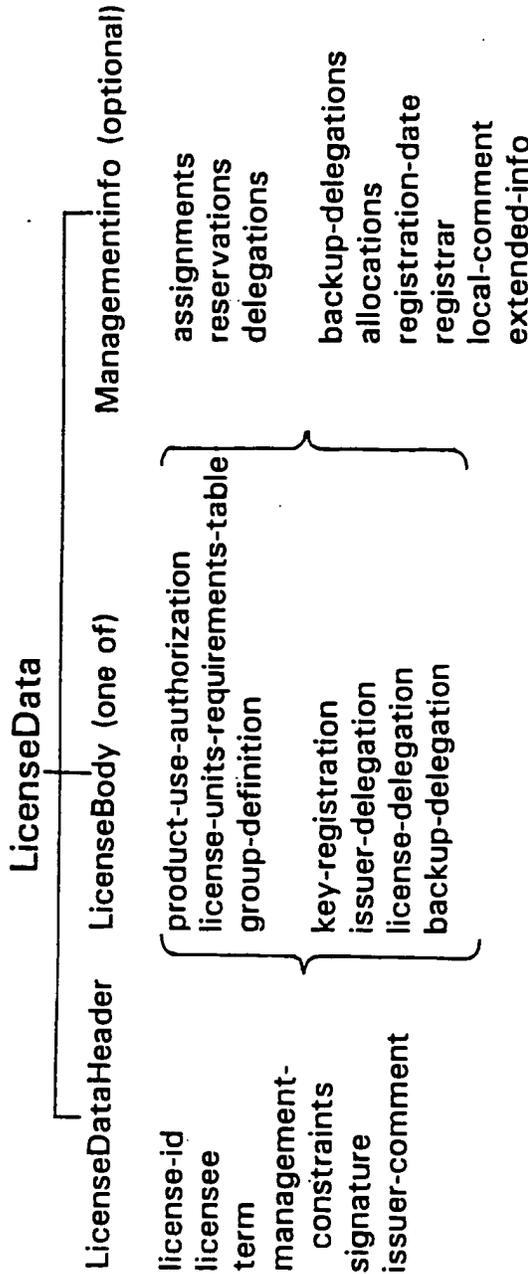


FIG. 16 License Data Structure

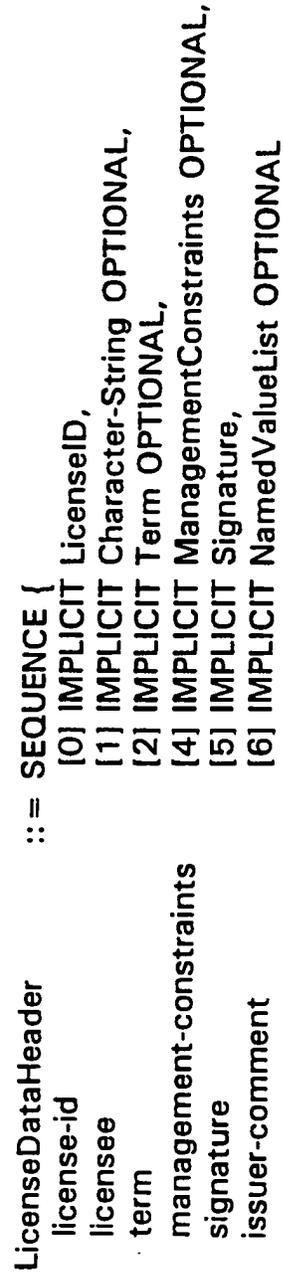


FIG. 17 License Data Header Syntax Diagram

```

ProductUseAuthorization ::= SEQUENCE {
  product-id          [0] IMPLICIT ProductID,
  units-granted       [1] IMPLICIT INTEGER,
  management-policy   [2] IMPLICIT ManagementPolicy,
  calling-authorizations [3] IMPLICIT SEQUENCE OF Member OPTIONAL,
  caller-authorizations [4] IMPLICIT SEQUENCE OF Member OPTIONAL,
  execution-constraints [5] IMPLICIT ExecutionConstraints OPTIONAL,
  product-token       [6] IMPLICIT NamedValueList OPTIONAL
}

```

FIG. 18 Product Use Authorization Syntax Diagram

```

LURT ::= SEQUENCE {
  lurt-name      [0] IMPLICIT Character-String,
  rows           [1] IMPLICIT RowList
}
RowList ::= SEQUENCE OF LurtRow

LurtRow ::= SEQUENCE {
  platform-id    [0] IMPLICIT Character-String,
  lurt-columns   [1] IMPLICIT SEQUENCE OF INTEGER
}

```

FIG. 19 License Unit Requirement Table Syntax Diagram

```

Example LURT ::= {
  lurt-name { Character-String "Example LURT" }
  rows {
    LurtRow {
      {Character-String "PC-0"}
      {{10} {230} {-1}}
    }
    LurtRow {
      {Character-String "PC-1"}
      {{12} {230} {-1}}
    }
    LurtRow {
      {Character-String "VAX 6210"}
      {{158} {300} {150}}
    }
  }
}

```

FIG. 20 Example Encoding of LURT

```

Group Definition      ::= SEQUENCE {
    group-name        [0] IMPLICIT Character-String,
    group-version     [1] IMPLICIT Version,
    group-release-date [2] IMPLICIT UTCTime,
    group-members     [3] IMPLICIT SEQUENCE OF Member
}
    
```

FIG. 21 Group Definition Syntax Diagram

```

KeyRegistration      ::= SEQUENCE {
    key-owner-name    [0] IMPLICIT Character-String,
    key-algorithm     [1] IMPLICIT Character-String,
    key-value         [2] IMPLICIT OCTET STRING
}
    
```

FIG. 22 Key Registration Syntax Diagram

SUBSTITUTE SHEET

```

IssuerDelegation
  delegated-issuer-name
  delegated-product-id
  delegated-units-granted
  template-authorization
  sub-license-permitted
 ::= SEQUENCE {
   [0] IMPLICIT Character-String,
   [1] IMPLICIT SEQUENCE OF Member,
   [2] IMPLICIT INTEGER OPTIONAL,
   [3] IMPLICIT ProductUseAuthorization OPTIONAL,
   [4] IMPLICIT BOOLEAN DEFAULT FALSE
 }

```

FIG. 23 Issuer Delegation Syntax Diagram

```

LicenseDelegation
  delegated-units
  delegated-distribution-control
  delegatee-execution-constraints
  assignment-list
  delegated-data
 ::= SEQUENCE {
   [0] IMPLICIT INTEGER OPTIONAL
   [1] IMPLICIT DistributionControl,
   [2] IMPLICIT ExecutionConstraints OPTIONAL,
   [3] IMPLICIT AssignmentList OPTIONAL,
   [4] IMPLICIT LicenseData OPTIONAL
 }

```

FIG. 24 License Delegation & Backup Delegation Syntax Diagrams

```

ManagementInfo
  assignments
  reservations
  delegations
  backup-delegations
  allocations
  registration-date
  registrar
  local-comment
  termination-date
  extended-info
  ::= SEQUENCE {
    [0] IMPLICIT AssignmentList OPTIONAL,
    [1] IMPLICIT AssignmentList OPTIONAL,
    [2] IMPLICIT DelegationList OPTIONAL,
    [3] IMPLICIT DelegationList OPTIONAL,
    [4] IMPLICIT AllocationList OPTIONAL,
    [5] IMPLICIT UTCTime,
    [6] IMPLICIT Context,
    [7] IMPLICIT NamedValueList OPTIONAL,
    [8] IMPLICIT UTCTime OPTIONAL,
    [9] IMPLICIT NamedValueList OPTIONAL
  }

```

FIG. 25 ManagementInfo Syntax Diagram

SUBSTITUTE SHEET

```

AllocationList ::= SEQUENCE OF Allocation
Allocation ::= SEQUENCE {
  allocation-context [0] IMPLICIT Context,
  allocation-lur [1] IMPLICIT INTEGER,
  allocation-group-id [2] IMPLICIT INTEGER OPTIONAL
}

```

FIG. 26 Allocation Syntax Diagram

```

AssignmentList ::= SEQUENCE OF Assignment
Assignment ::= SEQUENCE {
  assigned-units [0] IMPLICIT INTEGER,
  assignment-term [1] IMPLICIT Term,
  assignee [2] IMPLICIT Context
}

```

FIG. 27 Assignment Syntax Diagram

```

ContextList ::= SEQUENCE OF Context
Context ::= SEQUENCE OF SubContext
SubContext ::= SEQUENCE {
    sub-context-type [0] SubContextType,
    subcontext-value [1] ValueData
}
SubContextType ::= CHOICE {
    standard-subcontext-type [0] IMPLICIT INTEGER {
        network-subcontext(1),
        execution-domain-subcontext(2),
        login-domain-subcontext(3),
        node-subcontext(4),
        process-family-subcontext(5),
        process-id-subcontext(6),
        user-name-subcontext(7),
        product-name-subcontext(8),
        operating-system-subcontext(9),
        platform-id-subcontext(10)
    }
    private-subcontext [1] IMPLICIT INTEGER {first(0),last(255)}
}
    
```

FIG. 28 Context Syntax Diagram

SUBSTITUTE SHEET

FOOBAR V4.1 Allocated Units			Full Context Specifications
Units	Context Template		
	Node	User Name	
10	BLUE	WYMAN	ENET, AA_Cluster, BLUE, PID-1..., WYMAN
10	RED	OLSEN	ENET, BB_Cluster, RED, PID-1..., OLSEN
10	RED	WYMAN	ENET, BB_Cluster, RED, PID-2..., WYMAN
10	GREEN	WYMAN	ENET, AA_Cluster, GREEN, PID-1..., WYMAN
	GREEN	WYMAN	ENET, AA_Cluster, GREEN, PID-2..., WYMAN

FIG. 29 Only unique contexts require explicit unit allocations.

FOOBAR V4.1 Allocated Units		
Units	Context Template	Full Context Specifications
	Node	
10	BLUE	ENET, AA_Cluster, BLUE, PID-1..., WYMAN
10	RED	ENET, BB_Cluster, RED, PID-1..., OLSEN
	RED	ENET, BB_Cluster, RED, PID-2..., WYMAN
10	GREEN	ENET, AA_Cluster, GREEN, PID-1..., WYMAN
	GREEN	ENET, AA_Cluster, GREEN, PID-2..., WYMAN

FIG. 30 Modification of Context_Template impacts units requirements.

```

DistributionControl ::= SEQUENCE {
    distribution-method [0] IMPLICIT INTEGER {
        refresh-distribution(1),
        initial-distribution-only(2),
        manual-distribution(3)
    },
    current-start-date [1] IMPLICIT UTCTime OPTIONAL
    current-end-date [2] IMPLICIT UTCTime OPTIONAL,
    refresh-interval [3] IMPLICIT IntervalTime OPTIONAL,
    retry-interval [4] IMPLICIT IntervalTime OPTIONAL,
    maximum-retry-count [5] IMPLICIT INTEGER OPTIONAL,
    retries-attempted [6] IMPLICIT INTEGER OPTIONAL
}
    
```

FIG. 31 Distribution Control Syntax Diagram

```

ExecutionConstraints ::= SEQUENCE {
  operating-system      [0] IMPLICIT SEQUENCE OF Character-String OPTIONAL,
  execution-context    [1] IMPLICIT ContextList OPTIONAL,
  environment-list     [2] IMPLICIT SEQUENCE OF EnvironmentKind OPTIONAL
}
EnvironmentKind ::= INTEGER {
  batch(1),
  interactive(2),
  local(3),
  network(4),
  remote(5)
}

```

FIG. 32 Execution Constraints Syntax Diagram

```
LicenseID      ::= SEQUENCE {  
    issuer      [0] IMPLICIT Character-String,  
    serial-number [1] IMPLICIT Character-String,  
    amendment   [2] IMPLICIT INTEGER DEFAULT 0  
}
```

FIG. 33 License ID Syntax Diagram

```

LURDM ::= SEQUENCE {
    combination-permitted [0] IMPLICIT BOOLEAN DEFAULT TRUE,
    overdraft-limit [1] IMPLICIT INTEGER DEFAULT 0,
    overdraft-logging-required [2] IMPLICIT BOOLEAN DEFAULT FALSE,
    allocation-size [3] IMPLICIT INTEGER OPTIONAL,
    lurdm-kind [4] IMPLICIT INTEGER {
        lurt(1),
        constant(2),
        private-lurdm(3)
    },
    named-lurt-id [5] IMPLICIT Character-String OPTIONAL,
    lurdm-value [6] IMPLICIT INTEGER OPTIONAL,
    default-unit-requirement [7] IMPLICIT INTEGER OPTIONAL
}
    
```

FIG. 34 License Unit Requirements Determination Method Syntax Diagram

```

ManagementConstraints ::= SEQUENCE {
  management-context          [0] IMPLICIT ContextList OPTIONAL,
  management-scope           [1] IMPLICIT INTEGER {
    single-platform(1),
    management-domain(2),
    entire-network(3)
  } OPTIONAL,
  backup-permitted           [2] IMPLICIT BOOLEAN DEFAULT TRUE,
  delegation-permitted       [3] IMPLICIT BOOLEAN DEFAULT TRUE,
  maximum-delegation-period [4] IMPLICIT IntervalTime OPTIONAL
}

```

FIG. 35 Management Constraints Syntax Diagram

SUBSTITUTE SHEET

```

ManagementPolicy ::= SEQUENCE {
  style
    allocative(1),
    consumptive(2),
    private-style(3)
  context-template
  duration
    transaction(1),
    assignment(2),
    immediate(3)
  lur-determination-method
  allocation-sharing-limit
  reassignment-constraint
} OPTIONAL,
[1] IMPLICIT SEQUENCE OF SubcontextType
OPTIONAL,
[2] IMPLICIT INTEGER {
} OPTIONAL,
[3] IMPLICIT LURDM OPTIONAL,
[4] IMPLICIT INTEGER OPTIONAL,
[5] IMPLICIT IntervalTime OPTIONAL

```

FIG. 36 Management Policy Syntax Diagram

```

Member
member-product
member-signature
member-token
 ::= SEQUENCE {
    [0] IMPLICIT ProductID,
    [1] IMPLICIT Signature,
    [2] IMPLICIT NamedValueList OPTIONAL
 }

```

FIG. 37 Member Syntax Diagram

```

NamedValue
value-name
value-data
 ::= SEQUENCE {
    Character-String,
    ValueData
 }

ValueData
value-boolean
value-integer
value-text
value-general
value-list
 ::= CHOICE {
    [0] IMPLICIT BOOLEAN,
    [1] IMPLICIT INTEGER,
    [2] IMPLICIT SEQUENCE OF Character-String
    [3] IMPLICIT OCTET STRING,
    [4] IMPLICIT SEQUENCE OF ValueData
 }

```

```

NamedValueList
 ::= SEQUENCE OF NamedValue

```

FIG. 38 Named Value, Value Data & Named Value List Syntax Diagrams

```

ExampleList NamedValueList ::= {
  NamedValue {
    value-name {Character-String "Purchase Order"}
    value-data {INTEGER 154493}
  }
  NamedValue {
    value-name {Character-String "Telephone Support #"}
    value-data {Character-String { + 1 (999) 555-1234}
  }
}

```

FIG. 39 Named Value List Example

```

ProductID
producer
product-name
first-version
last-version
first-release-date
last-release-date
} ::= SEQUENCE {
  [0] IMPLICIT Character-String,
  [1] IMPLICIT Character-String,
  [2] IMPLICIT Version OPTIONAL,
  [3] IMPLICIT Version OPTIONAL,
  [4] IMPLICIT UTCTime OPTIONAL,
  [5] IMPLICIT UTCTime OPTIONAL
}

```

FIG. 40 Product ID Syntax Diagram

```

Signature
signature-algorithm
signature-parameters
signature-value
 ::= SEQUENCE {
      [0] IMPLICIT Character-String,
      [1] IMPLICIT NamedValueList OPTIONAL,
      [2] IMPLICIT OCTET STRING
    }

```

FIG. 41 Signature Syntax Diagram

```

Term
start-date
end-date
 ::= SEQUENCE {
      [0] IMPLICIT UTCTime OPTIONAL,
      [1] IMPLICIT UTCTime OPTIONAL,
    }

```

FIG. 42 Term Syntax Diagram

```

Version
  part-1
  part-2
  part-3
  part-4
 ::= SEQUENCE {
   [0] IMPLICIT INTEGER,
   [1] IMPLICIT INTEGER DEFAULT 0,
   [2] IMPLICIT INTEGER DEFAULT 0,
   [3] IMPLICIT INTEGER DEFAULT 0
 }
    
```

FIG. 43

Attributes Specific to Filter				
Attribute	Value Syntax	Value Length	Value Number	Value Initially
Filter Items	Object(Filter Item)	-	0 or more	-
Filters	Object(Filter)	-	0 or more	-
Filter Type	Enum(Filter Type)	-	1	-

FIG. 44

Attributes Specific to Filter						
Attribute	Value Syntax	Value Length	Value Number	Value Initially		
Filter Item Type	Enum(Filter Item Type)	-	1	-		
Attribute Type	Type	-	1	-		
Match Value	any	-	0-1	-		
Filters	Object(Filter)	-	0-1	-		
Initial Substring	String(*)	1 or more	0-1	-		
Substring	String(*)	1 or more	0 or more	-		
Final Substring	String(*)	1 or more	0-1 or more	-		
License Request	Object(License Request)	-	0-1	-		

FIG. 45

```

Filter {
  Filter-Type AND
  Filter-Item {
    Filter-Item-Type SELECT
    Attribute-Type Product-Use-Authorization
    Filter {
      Filter-Type AND
      Filter-Item{
        Filter-Item-Type SELECT
        Attribute-Type Calling-Authorization
        Filter{
          Filter-Type AND
          Filter-Item {
            Filter-Item-Type EQUALITY
            Attribute-Type Producer
            Match-Value "Digital"
          }
          Filter-Item {
            Filter-Item-Type EQUALITY
            Attribute-Type Producer
            Match-Value "Amazing Database"
          }
        }
      }
    }
  }
  Filter-Item {
    Filter-Item-Type EQUALITY
    Attribute-Type Producer
    Match-Value "Digital"
  }
  Filter-Item{
    Filter-Item-Type EQUALITY
    Attribute-Type Issuer
    Match-Value "Digital"
  }
  Filter-Item {
    Filter-Item-Type EQUALITY
    Attribute-Type Product-Name
    Match-Value "Amazing Graphics System"
  }
}

```

FIG. 46 Example Filter Value Notation

INTERNATIONAL SEARCH REPORT

DCT/115 92/03812

International Application No

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) ⁶		
According to International Patent Classification (IPC) or to both National Classification and IPC Int.Cl. 5 G06F1/00		
II. FIELDS SEARCHED		
Minimum Documentation Searched ⁷		
Classification System	Classification Symbols	
Int.Cl. 5	G06F	
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched ⁸		
III. DOCUMENTS CONSIDERED TO BE RELEVANT⁹		
Category ⁹	Citation of Document, ¹¹ with indication, where appropriate, of the relevant passages ¹²	Relevant to Claim No. ¹³
Y	EP,A,0 332 304 (DIGITAL EQUIPMENT CORPORATION) 13 September 1989 cited in the application	1-3, 6-19, 22, 24, 26-29, 31-33, 35-37, 39 43-45
Y	see figure 1 cited in the application	5, 15, 21, 25, 30
A	see column 3, line 31 - column 7, line 55 ----- -/--	
<p>⁹ Special categories of cited documents : ¹⁰</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&" document member of the same patent family</p>		
IV. CERTIFICATION		
Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report	
2 09 SEPTEMBER 1992	17. 09. 92	
International Searching Authority	Signature of Authorized Officer	
EUROPEAN PATENT OFFICE	WEISS P.	

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)		
Category °	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.
Y	IBM TECHNICAL DISCLOSURE BULLETIN. vol. 31, no. 8, 1 January 1989, NEW YORK US pages 195 - 198; 'METHOD FOR MANAGING CLIENT/SERVER RELATIONSHIP IN THE AIX OPERATING SYSTEM'	1-3, 6-19, 22, 24, 26-29, 31-33, 35-37, 39 43-45
Y A	see the whole document	21

**ANNEX TO THE INTERNATIONAL SEARCH REPORT
ON INTERNATIONAL PATENT APPLICATION NO.**

US 9203812
SA 60557

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The members are as contained in the European Patent Office EDP file on The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information. 09/09/92

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-0332304	13-09-89	US-A- 4937863 JP-A- 2014321	26-06-90 18-01-90

EPO FORM P4679

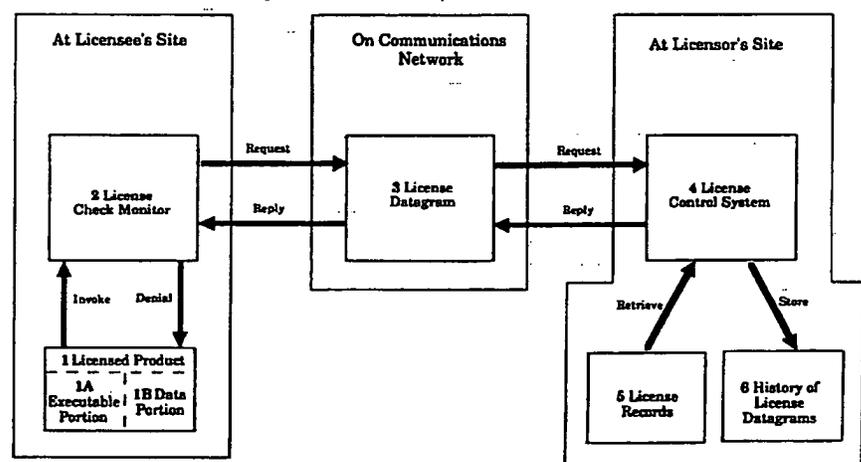
For more details about this annex : see Official Journal of the European Patent Office, No. 12/82



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁵ : G06F 11/34, H04L 9/00</p>	<p>A1</p>	<p>(11) International Publication Number: WO 93/01550 (43) International Publication Date: 21 January 1993 (21.01.93)</p>
<p>(21) International Application Number: PCT/US92/05387 (22) International Filing Date: 30 June 1992 (30.06.92) (30) Priority data: 724,180 1 July 1991 (01.07.91) US 907,934 29 June 1992 (29.06.92) US (71) Applicant: INFOLOGIC SOFTWARE, INC. [US/US]; 1223 Peoples Avenue, Suite 5405, Troy, NY 12180 (US). (72) Inventor: GRISWOLD, Gary, N. ; 1937 Regent Street, Schenectady, NY 12309 (US). (74) Agents: LAZAR, Dale, S. et al. ; Cushman, Darby & Cush- man, Ninth Floor, 1100 New York Avenue, N.W., Wash- ington, DC 20005-3918 (US).</p>		<p>(81) Designated States: AT, AU, BB, BG, BR, CA, CH, CS, DE, DK, ES, FI, GB, HU, JP, KP, KR, LK, LU, MG, MN, MW, NL, NO, PL, RO, RU, SD, SE, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IT, LU, MC, NL, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, SN, TD, TG).</p> <p>Published <i>With international search report.</i></p>

(54) Title: LICENSE MANAGEMENT SYSTEM AND METHOD



(57) Abstract

A license management system and method for recording (6) the use of licensed product (1), and for controlling (4) its use. A licensed product invokes a license check monitor (2) at regular time intervals. The monitor generates request datagrams (3) which identify the licensee and the product and sends the request datagrams over a communications facility to a license control system (4). The license control system maintains a record (6) of the received datagrams, and compares the received datagrams to data stored in its licensee database (5). Consequently, the license control system (4) transmits reply datagrams with either a denial or an approval message. The monitor (2) generates its own denial message if its request datagrams are unanswered after a predetermined interval of time. The datagrams are counted at the control system to provide billing information.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FI	Finland	MI	Mali
AU	Australia	FR	France	MN	Mongolia
BB	Barbados	GA	Gabon	MR	Mauritania
BE	Belgium	GB	United Kingdom	MW	Malawi
BF	Burkina Faso	GN	Guinea	NL	Netherlands
BG	Bulgaria	GR	Greece	NO	Norway
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IE	Ireland	RO	Romania
CA	Canada	IT	Italy	RU	Russian Federation
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland			SN	Senegal
CI	Côte d'Ivoire	KR	Republic of Korea	SU	Soviet Union
CM	Cameroon	LI	Liechtenstein	TD	Chad
CS	Czechoslovakia	LK	Sri Lanka	TC	Togo
DE	Germany	LU	Luxembourg	US	United States of America
DK	Denmark	MC	Monaco		
ES	Spain	MG	Madagascar		

- 1 -

LICENSE MANAGEMENT SYSTEM AND METHOD

BACKGROUNDField of the Invention

5 The present invention generally relates to systems for managing licenses of products such as computer software, video games, CD-ROM information, movies and other video products, music and other audio products, multimedia products, and other systems for up-to-date recording of actual usage of such a
10 licensed product to enable efficient billing therefor.

Description of Related Art

Licenses for information products such as computer software, music, video products and the like usually provide licensees with limited rights. The
15 licenses may restrict sites of use, duration of use, or number of concurrent uses of the products. The licenses also may limit the use of the products depending on currentness of licensee's payments. However, enforcing the conditions of the licenses is
20 difficult, because, in general, the licensed products may be easily copied or "pirated" and used without the licensor's knowledge.

Compliance with limited license rights has been encouraged with copy protection. Known methods of
25 computer software copy protection include putting a

SUBSTITUTE SHEET

physical hole or mark on the diskette containing a product, or placing data on the diskette in a location where no data is expected. A disk with an illegally copied software product usually would not contain the marks. At the beginning of its operation, a copy-protected, but illegally copied software product would search its own diskette for the marks. Upon failing to detect the marks, the software would abort from its normal procedures.

10 Most software products sold today do not have such copy protection, partly because copy protection renders legitimate duplication of copy protected software difficult, but not impossible. Copy protection frustrates the making of legitimate copies, while not eliminating unauthorized copying. Many software publishers have experienced higher sales by eliminating copy protection schemes.

Another method for enforcing limited licensing rights of computer software is described in U.S. patent No. 4,932,054 to Chou. Chou describes a "coded filter" hardware device which is plugged into a port of a computer. The "coded filter" outputs an authorization control code when a predetermined control code is sent to it. The licensed software functions properly only if the "coded filter" transmits the correct authorization control code to the software.

While devices such as described by Chou have existed for several years, they have not been well accepted by the market. Since the device is attached to the outside of a computer, it can easily be lost or stolen, preventing the use of licensed software. In addition, if a licensee purchased a number of software

products, each of which used Chou's protection scheme, the licensee would collect a stack of "coded filters."

Hershey, in U.S. patent No. 4,924,378, describes a method for limiting the number of concurrent uses of a licensed software product. Each workstation of a network has a license storage area in its local memory. License Management System (LMS) daemons are provided in the network in a number corresponding to the permissible number of concurrent uses of the software product. To use the software, a work station stores a daemon in its license storage area. If all daemons are in use, no further work stations may use the software.

Robert et al., in U.S. patent No. 4,937,863, describe a similar invention. This invention includes a license management facility which accesses a database of license information related to licensed computer software programs. When a user attempts to use a licensed program, the license management facility first checks the database. Access to the licensed product is prevented if licensing conditions related to the product are not satisfied (e.g., expiration of licensing dates, etc).

While the Robert et al. and Hershey patents show effective techniques for controlling licensed computer software, each also reveals components that cannot be easily managed by an average user. A system manager, or someone with special access privileges to the internals of a machine, must install the licensed software. This hinders the distribution of the software.

Licensable products other than computer software have not generally been copy-protected. For example,

video tapes can be easily copied by anyone with two VCR machines, and audio tapes and music CDs can be easily copied to tape. Computer CD-ROMs can be copied to magnetic disk; however, their large information storage capacity relative to that of magnetic disks makes this a very expensive proposition. The introduction of digital audio tape is being delayed, because some view its ability to easily produce very high quality copies as a threat to music royalties.

10 Hellman, in U.S. patent No. 4,658,093, describes means to bill by usage. This is accomplished via communication of an encrypted authorization code from a licensor to a base unit at the licensee's site. The encrypted authorization code contains information
15 related to an identification of the base unit, a number of uses requested, and a random or non-repeating number; however, implementation of Hellman's scheme requires a "base unit", such as a computer, video game unit, record player, video recorder, or
20 video disk player, with a unique identification number. The requirement is difficult to satisfy, because, at the present, only a fraction of such systems on the market have an internally readable serial number for identification. In addition,
25 vendors of these systems provide no guarantees for the uniqueness of any given device's serial number. Furthermore, an internal serial number can change when hardware maintenance is performed on the device. Also, Hellman's approach requires that an identical
30 copy of each software product be stored at the authorization site. These copies are used in the generation of unique keys. The unstated assumption that all copies of a specific version of a software

product are identical is unrealistic. Minor bug fixes to software are often made without generating a new version of the product. Also, some software products, such as those which run on Macintosh computers, are self-modifying.

While Hellman's invention counts each use of the software, it does not monitor the duration of use. Thus, Hellman's system would not be able to bill for extensive use of licensed software if the software were continuously operated. Finally, while Hellman suggests the inclusion of an automated communication system as part of his invention, he does not disclose how this communication system could be implemented. Instead, he mentions non-automated use of telephone and mail. In summary, Hellman's patent is an interesting discussion of cryptographic techniques, but it does not provide a practical, real-world implementation of those techniques.

Shear, in U.S. Patent No. 4,977,594, describes a system and method to meter usage of distributed databases such as CD-ROM systems. The method describes a hardware module which must be part of the computer used to access the distributed database. This module retains records of the information viewed. Once the module storage is filled, the module must be removed and delivered to someone who will charge for the usage recorded therein and set the module back to zero usage. Like Hellman's method, this method requires a hardware module which must be incorporated within the computer so the system can control user access. No database publisher will be able to use this method until there are a very large number of units containing such modules. Hardware manufacturers

will be hesitant to include the module in the design of their computers until there is sufficient demand from customers or publishers for this system. The method and apparatus according to the present invention can be implemented entirely in software and hence does not require special, dedicated computer subsystems.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a license management system and method which can ensure that a licensed product is used only on machines under which it is licensed.

It is another object of the present invention to provide a license management system and method which may terminate access to a licensed product once its license has expired.

It is yet another object of the present invention to provide a license management system and method which may terminate access to a licensed product when payment for a license is overdue.

It is a further object of the present invention to provide a license management system and method which can limit the number of concurrent uses of a licensed product.

It is yet another object of the present invention to provide a license management system and method which can bill licensees for the duration of actual usage of a licensed product.

The present invention provides an advantageous feature of quickly and effectively implementing license agreements between a licensor and licensee.

The present invention provides another advantageous feature of allowing logic used to control licenses to be easily changed.

5 The present invention provides yet another advantageous feature of detecting, at the licensor's site, many types of attempts to alter the license management system.

10 The present invention provides a further advantageous feature of permitting anyone without special access privileges to install a licensed product.

15 In the present invention, a licensed product generates request "datagrams," messages transmitted over a communications network. The request datagrams are sent to the licensor's site. At the licensor's site the datagram is compared to information stored in a license database. After the comparison, a reply datagram is sent to the licensee. Upon receiving the reply datagram, the licensed product reacts in
20 accordance with the instructions therewithin. For example if a reply datagram contained a "denial," the licensed product would display an appropriate message to the user and then suspend further execution of its programs.

25 In the present invention, the licensed product is implemented on a network node attached to a communications network that includes the licensor. The network node may be a computer, a CD-ROM player, a tele-computer or other multimedia machine, or any
30 other appropriate device. The node may also be an intelligent type of consumer electronic device used for presenting information, such as an intelligent television, VCR, videodisk player, music CD player,

audio tape player, telephone or other similar device. Further, the communications network may be any two-way network such as a computer network, telephone network, a cellular telephone network or other
5 wireless network, a two-way cable TV network, or any other equivalent system.

Should the user detach the node from the network, the licensed product will fail to receive reply datagrams. Upon several failures to receive reply
10 datagrams, the licensed product will generate its own denial.

After a request datagram has been sent out, a user may be permitted to use the licensed product for a limited duration. This feature may be necessary
15 because of the delays in network communications. When networks are sufficiently fast, use of a licensed product can be postponed until the reply datagram is received.

In the preferred embodiment of the present
20 invention, licensees' network addresses are used to identify the licensees. Other embodiments may use a licensed product serial number or hardware serial numbers for the identification.

A licensed product as in the present invention
25 generates a request datagram after each period of product use. The number of request datagrams received by the licensor can be used to bill the licensee. For example, if datagrams are sent after every hour of product use, the licensee will be billed for the
30 amount equal to the number of request datagrams received by the licensor multiplied by the hourly rate.

The embodiments of the present invention may incorporate a query system at a licensor's site for reporting on problem datagrams. This would allow the licensors to take appropriate actions in accordance with problems associated with each datagram.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects and advantages of this invention will become more apparent and more readily appreciated from the following detailed description of the presently preferred exemplary embodiment of the invention, taken in conjunction with the accompanying drawings, of which:

FIGURE 1 is a general block diagram of the preferred exemplary embodiment of the present invention;

FIGURE 2 shows representative diagrams of the contents and formats of data at licensee's site, contained in datagrams, and at licensor's site;

FIGURE 3 illustrates a sequence of representative operations executed at the licensee's site and at the licensor's site, together with required inputs for the execution of the operations and with outputs produced therefrom;

FIGURE 4 illustrates a sequence of representative operations to send a request datagram, together with required inputs for the execution of the operations and with outputs produced therefrom;

FIGURE 5 illustrates a sequence of representative operations when a reply datagram is overdue, together with required inputs for the execution of the operations and with outputs produced therefrom;

FIGURE 6 shows a sequence of representative operations to process a reply datagram, together with required inputs for the execution of the operations and with outputs produced therefrom;

5 FIGURE 7 shows a sequence of representative operations to generate an authorization code, together with required inputs for the execution of the operations and with outputs produced therefrom; and

10 FIGURE 8 shows a sequence of representative operations to send a reply datagram, together with required inputs for the execution of the operations and with outputs produced therefrom.

DETAILED DESCRIPTION OF THE
PRESENTLY PREFERRED EXEMPLARY EMBODIMENT

15 As shown in FIGURE 1, a licensed product 1 is located at a licensee's site. Product 1 may include a data portion 1B and a functional portion 1A such as computer software product or any other kind of information product used to control use of data
20 portion 1B. If data portion 1B is CD-ROM database information, functional portion 1A should enable the licensee to search indexes and display text. If data portion 1B is video information, functional portion 1A should control the display of the video information.
25 For audio information, functional portion 1A should play the audio information. If data portion 1B is an electronic book, functional portion 1A should display and turn pages. The above examples show some of the ways functional portion 1A can control data portion
30 1B; however, they are hardly exhaustive.

By including in product 1 both information and software which controls the information, product 1 is

an executable product. Non-software information in product 1 is preferably encrypted so that it cannot be easily extracted from the product.

License check monitor 2 sends license datagrams 3 to the licensor and also receives license datagrams 3 from the licensor. License check monitor 2 also prevents further use of product 1 when a datagram 3 containing a "denial" message is received.

License datagrams 3 are messages that describe information related to the use of licensed product 1. Datagrams 3 are sent over a communications network between the licensee and licensor. Initially, the licensee sends a request datagram 3 over the network to the licensor. The licensor then returns a reply datagram containing either an approval or denial. It is also possible to implement the present invention by having the licensor transmit a reply datagram only for approvals.

At the licensor's site, license control system 4 makes licensing decisions by comparing request datagram 3 with license records 5. After the comparison, control system 4 stores information related to request datagram 3 into history of license datagram record 6. It is noted that request datagrams 3 are periodically sent while product 1 is in use. Thus, the history of license datagrams in record 6 provides means for measuring the duration of use of product 1.

Representations of data and records stored at the licensee's site, contained in datagrams, and stored at the licensor's site are illustrated in FIGURE 2. At the licensee's site, network service 7, which handles delivery and transmission of datagrams 3, supplies

network address 8. It is by this address that license control system 4 identifies a location of use of product 1.

5 Licensed product record 9 is contained within monitor 2. Within the license product record 9 is an identification record 10, which contains the following two items: licensor's network address 11, and product model number 12 that identifies product 1. When a
10 licensor has only one product, or uses different licensor network addresses 11 for each product, product model number 12 may not be needed.

Datagram sent record 13 stores information about the last sent datagram 3. It includes a datagram number 14, which uniquely identifies the last
15 transmitted datagram 3, and the date and time 15 when the last datagram 3 was sent from the licensee's site.

Licensed product record 9 also contains control parameters record 16, which is used for controlling the timing of key events in the communication of
20 license check monitor 2 with license control system 4. Send interval 17 specifies a time interval between each transmission of a new datagram 3 from the licensee to the licensor.

Wait interval 18 is the length of time that
25 monitor 2 waits to receive a reply datagram 3 before resending the same request datagram 3. The duration of this interval depends on the speed of the communications network being used to deliver datagrams 3.

30 Disconnect allowed interval 19 is the duration of time that monitor 2 allows product 1 to be used without a reply datagram 3 from the licensor. The duration of this interval depends on the reliability

of the communications network. The interval must be long enough to take into consideration network downtime. For example, suppose a message was sent from the licensor and the network went down just afterwards. Disconnect allowed interval 19 should be long enough to allow the network to resume its normal operation and successfully deliver datagrams 3 from the licensor; otherwise, the licensee would be forced to stop using product 1 until the network was operational.

License datagram 3 contains header 20. Header 20 is used during execution of low level communication protocols within the network. Source network address 21 is the network address from where datagram 3 is sent. Destination network address 22 is the network address to where datagram 3 is sent. Additional data may be included in header 20 if required by low level protocols used in delivering datagrams 3.

Data 23, a part of datagram 3, conveys a message, and contains a number of fields. Product model number 24 and datagram number 25 identify product 1 and datagram 3, respectively. It is noted that retransmitted datagrams have an identical datagram number. Duplicate datagrams must be identified at a licensor's site so that they do not all contribute in billing a licensee.

Each datagram number 25 is unique for each request datagram 3 transmitted from the licensee, except for retransmitted datagrams. This allows a reply datagram 3 received by a licensee to be verified as an actual reply to a request datagram 3 from that licensee, as explained below.

Number of processes running 26 is the number of concurrent uses of product 1 at the time datagram 3 is sent. Authorization code 27 is used on reply datagrams 3 to indicate an approval or a denial. 5 Message text 28 contains a message which will be displayed to the user upon a denial.

License database 29 at the licensor's site holds records of information about customers, licenses, and license usage. The types of information within 10 license database 29 of the present embodiment are shown in FIGURE 2. However, a specific license management system may require its license database to hold types of information other than those in FIGURE 2. For example, licensee name and address may be 15 incorporated as a part of a license database 29.

License record 5 contains information on licenses. Licensee network address 30 identifies a precise network node which is licensed to use product 1. If request datagrams are received which do not 20 originate from known licensee network addresses 30, reply datagrams containing denial messages are transmitted. Product model number 31 is the model number of a licensed product. Termination date 32 is the expiration date of a license. When the license of 25 a product is issued for an unlimited duration, termination date 32 should reflect a date very far into the future, relative to the licensing date.

The present embodiment allows licenses to be paid for in a lease-like or rental fashion. If a licensee 30 were to rent or lease product 1, paid through date 33 would reflect the date through which the licensee has paid for using the product. Grace period 34 is the time interval for which the licensee is allowed to be

delinquent before services are disconnected. Grace period 34 would reflect a very large time interval if the license is not of a lease-like or rental type. When the license provides for a limit on the number of concurrent uses of a product 1, number of processes licensed 35 contains the limiting number. When the license does not provide for such a limit, number of processes 35 should be a very large number.

History of license datagrams 6 is an archive of datagrams 3 received from the licensee.

FIGURE 3 illustrates operations executed at the licensee's site and at the licensor's site. An overview of the processing at the licensee's site is described by steps 101.0 to 106.0, and an overview of the processing at the licensor's site is described by steps 107.0 to 110.0.

At the licensee's site, at step 101.0, product 1 invokes monitor 2. This is accomplished by first establishing monitor 2 as a handler for a timer expiration interrupt signal and for received datagrams 3. Next, a timer is set with a very short time to cause an initial call to monitor 2. At step 102.0, monitor 2 computes a time 36 since the last datagram was sent by determining the difference between the current date and sent time and date and time 15 that a datagram was last sent from the licensee's site. When product 1 commences execution, datagram sent date and time 15 is set to "null." Thus, time since send 36 is very large at the beginning of the monitor's execution. At step 103.0, time since send 36 is compared to send interval 17. If time since send 36 is greater than send interval 17, then a request datagram is transmitted, per the steps described in

FIGURE 4. Step 104.0 first checks if a reply to the last datagram has arrived and if wait interval 18 has expired. If a reply has not arrived and the wait interval has expired, steps 104.1-104.3 (FIGURE 5) are executed. Step 105.0 processes authorization code 27 in a reply when the reply is received, in accordance with steps 105.1 to 105.5 (FIGURE 6). At step 106.0, product 1 resumes normal execution of its programs until the next interrupt signal is generated.

At the licensor's site, license control system 4 receives and processes datagram 3, in accordance with steps 107.0 to 110.0. Step 107.0 receives request datagram 3. Step 108.0 generates authorization code 27, per steps 108.1 to 108.8 (FIGURE 7). Step 109.0 creates reply datagram 3 and transmits the datagram to the licensee via steps 109.1 to 109.5 (FIGURE 8).

FIGURE 4 shows the procedure which monitor 2 follows for sending request datagram 3 to the licensor. Step 103.1 sets source network address 21 in datagram 3 to the network address 8 of the licensee's location on the network. Step 103.2 sets destination network address 22 to licensor's network address 11. Step 103.3 encrypts product model number 12 for datagram 3. Step 103.4 assigns a unique number to datagram 3, encrypts the number, and stores it as datagram number 14. This number is altered when an entirely new datagram 3 is sent. Datagrams which are retransmitted have the same datagram number 25 as the original. As already explained, this allows license control system 4 to identify duplicate datagrams.

Step 103.5 counts the number of processes using product 1, currently running, encrypts the count, and stores the encryption as the number of processes

running 26. In the UNIX operating system, this procedure could be performed using the command "ps" to obtain a list of current processes, the command "grep" to extract the processes of product 1, and "wc" to count the number of processes. Step 103.6 sets authorization code 27 to number 255 and encrypts the number.

Number 255 indicates that datagram 3 is a request for authorization. Such an indication is needed to guard the present system against the following steps for circumventing the present invention: intercepting outgoing datagrams; and inputting the intercepted datagrams to monitor 2.

Step 103.7 stores the current date and time as sent date & time 15. This date is needed to compute when to send the next datagram 3. Step 103.8 assigns a value to send interval 17, which sets an alarm for invoking monitor 2 to send the next datagram 3. Step 103.9 sends datagram 3.

In the present embodiment a datagram is transmitted via a connectionless datagram service. Methods for transmission are well documented for some networking systems. For example, TCP/IP (Transport Control Protocol/Internet Protocol) includes a connectionless protocol called UDP (User Datagram Protocol). A method for sending a datagram using UDP protocol from a SUN Microsystem computer is documented in a SUN manual titled, Network Programming Guide, in section 9 titled "Transport Level Interface Programming."

Step 103.10 sets another alarm using wait interval 18 for retransmitting datagram 3, if no reply datagram has been received. The alarm causes monitor

2 to be invoked for checking whether a reply datagram
3 has been received. Monitor 2 will transmit a
duplicate of the previously transmitted datagram, if
no reply has been received. After the execution of
5 step 103.10, "Send License Datagram" procedure returns
system control to step 104.0 in FIGURE 3.

FIGURE 5 shows the operation of the "Reply
Datagram is Overdue" procedure. Step 104.1 compares
time since the last datagram was sent 36 to disconnect
10 allowed interval 19, which, as described above, is the
interval that product 1 is allowed to operate even if
a reply is overdue. If time since send 36 is smaller
than disconnect allowed interval 19, datagram 3 is
retransmitted via executing step 103.9 in FIGURE 4.
15 Step 104.2 "disconnects" product 1 from further
service, if time since send 36 is greater than
disconnect allowed interval 19.

Step 104.2 comprises a sequence of sub-steps
104.2.1-104.2.3. Step 104.2.1 assigns number 5 to
20 authorization code 27 in the current datagram being
processed. Value 5 is interpreted by monitor 2 as a
denial. Step 104.2.2 sets message text 28 to the
following: "A reply from licensor to numerous
authorization requests was never received. This
25 product must be connected to a communications network
in order to function." Step 104.2.3 transfers system
control to step 105.3 in FIGURE 6. Step 105.3
processes the current denial datagram 3 as if it were
just received.

30 Through the execution of steps 104.1-104.3, the
present system permits the use of product 1 for a
prescribed period of time. After the prescribed

period of time has elapsed, the present system generates a denial.

FIGURE 6 illustrates the steps which monitor 2 follows in processing a reply datagram 3. Step 105.1
5 decrypts all encrypted data in the received datagram. Step 105.2 compares datagram number 25 with datagram number 14 associated with the last datagram. If datagram number 25 is not equal to datagram number 14, step 105.2 ignores the current datagram and transfers
10 procedural control to step 103.9 (FIGURE 4) in order to resend the last transmitted datagram. After disconnect allowed interval 19 elapses, monitor 2 generates a denial.

In essence, step 105.2 guards against the
15 circumvention of the present invention via: (1) intercepting a reply datagram 3 (from the licensor) containing an approval (2) storing the reply datagram 3; and (3) inputting the stored datagram to monitor 2.

If the execution of step 105.2 does not transfer
20 its procedural control to step 105.3, and if authorization control 27 is not zero (indicating an unqualified authorization has not been received), step 105.3 processes authorization code 27 via steps 105.3.1 to 105.3.3. Step 105.3.1 retrieves message
25 text 28 from datagram 3. If message text 28 is null, then the current datagram 3 is ignored, and monitor 2 resends the last transmitted datagram 3. Step 105.3.1 further protects the present system from attempts to generate fake datagrams and to feed the fake datagrams
30 to monitor 2 by checking for a proper authorization code of zero.

If message text 28 is not null, step 105.3.2 presents the message 28 to the user on an output

device such as a CRT screen. Step 105.3.3 terminates the current use of product 1. This step may be implemented by subroutine or function call to a simple exit that saves any current user data to a file.

5 Alternatively, product 1 may be designed so that, upon being directed to terminate further execution, it first gives the user an opportunity to save their data.

If authorization code 27 is zero, step 105.4
10 allows further use of product 1. Step 105.5 returns procedural control to 106.0 on FIGURE 3.

FIGURE 7 shows a sequence of operations within the "Generate Authorization Code" procedure. The procedure produces appropriate authorization code 27
15 when a request datagram 3 is received at the licensor's site.

Step 108.1 decrypts all encrypted data in the received datagram 3. Using source network address 21 and product model number 24 in the datagram 3, step
20 108.2 searches the license database 29 for matching licensee network address 30 and product model number 31. If license database 29 does not contain a record of product model number 24 of the product 1 being licensed to the licensee, step 108.3 sets
25 authorization code 27 of its reply datagram 3 to 1 (i.e., the sending node is not a registered address) and authorization is denied.

Step 108.3 prevents copies of product 1 from being installed on multiple nodes independently of
30 whether they are within or outside the licensee's organization. Step 108.3 also prevents the licensee from transporting product 1 from one node to another node without the licensor's approval. This is

important because the two nodes may have different processing capacities, and they may be billable at different rates.

5 If the date a request datagram is received is later than license termination date 32, step 108.4 sets authorization code 27 to number 2 (i. e., license has expired). Step 108.4 allows the licensor to fix licensing periods, or to determine free trial periods for the use of the product. The licensing period may
10 be extended by resetting license termination date 32 at the licensor's site.

If the date when the datagram is received is later than the paid through date 33 as extended by the grace period 34, step 108.5 sets authorization code 27
15 to 3 (i.e., payment is past due).

If the number of processes running 26 exceeds a licensed number of concurrent uses of product 1 (at a particular node), then step 108.6 sets authorization code 27 to 4 (i.e. concurrent process usage limit is exceeded).
20

Step 108.7 sets authorization code 27 to 0 indicating processing can continue. It is noted that steps 108.3-108.7 are a part of a

25 IF (x1) then (y1)
ELSE if (x2) then (y2)
ELSE if (x3) then (y3) ...

statement of a procedure (e. g., FORTRAN, PASCAL, C, etc). Thus, only one of the steps 108.3-108.7 is executed. Step 108.7 sets authorization code 27 to 0
30 (indicating approval of further use) only if steps 108.3-108.6 do not execute the THEN portion of each step. Step 108.7 also stores the received datagram 3 in history of license datagrams 6.

Step 108.8 is the last of authorization processing rules 108.1-108.7. After the execution of steps 108.3-108.7, step 108.8 returns procedural control to step 109.0 in FIGURE 3.

5 FIGURE 8 illustrates the steps which license control system 4 follows to send reply datagram 3 to the licensee.

Step 109.1 encrypts authorization code 27 and writes the encrypted code into datagram 3. Next, step 10 109.2 writes message text 28 corresponding to authorization code 27 into datagram 3.

Step 109.2 may be replaced with the following method for relaying proper messages to a product user. Proper messages corresponding to each authorization 15 code is stored in monitor 2 at each licensee's site. Upon reception of a reply datagram 3, monitor 2 would locate within itself the proper message corresponding to the authorization code, and use the message for various purposes. This method would reduce the size 20 of reply datagrams 3. However, if the licensor wanted to implement new denial codes, each product would need to somehow incorporate the new message associated with the new denial code into itself. The list of messages, one of which may be written as message text 25 28, are as follows:

AUTHORIZATION
CODE

TEXT MESSAGE

30	1	This product is not licensed to run at this location. Please contact the licensor to either license this product, or move an existing license of your organization to this location. Use of this product at this
----	---	--

location is discontinued until this problem is resolved.

2
5 Your license on this product has expired. Please contact licensor in order to have your license extended. Use of this product is discontinued until this problem is resolved.

10 3 Payment on this licensed product is over due and past your grace period. Please have your accounting department send payment in order to continue your license. Use of this product is discontinued until this problem is resolved.

20 4 Your current use of this licensed product exceeds limits for the number of uses your organization has licensed. Please try again later.

25 5 A reply from licensor to numerous authorization requests was never received. This product must be connected to a communications network in order to function.

0 Authorization is OK. There is no message.

30 Step 109.3 swaps source network address 21 and destination network address 22. Step 109.4 transmits datagram 3 back to monitor 2.

35 At step 109.5, a communications network delivers datagram 3 to monitor 2. Subsequently, procedural control returns to step 107.0 in FIGURE 3 to process the next datagram 3.

Although only a few exemplary embodiments of this invention have been described in detail above, those skilled in the art will readily appreciate that many

modifications are possible in the preferred embodiments without materially departing from the novel teachings and advantages of this invention. For example, product 1 was described as sometimes
5 consisting of information as well as software which controls the information. This approach provides the greatest flexibility, but it is also possible to include the software which controls the information in the networked machine at the licensee's site. In this
10 case, product 1 is split, with part of it on media and part on the licensee's machine. By doing this, some space can be saved on the media containing product 1, but the capabilities of these products will be limited by the standard functions available on these machines.

15 Also, the presently described embodiment includes a product 1 which is at the licensee's site. This implies that product 1 is on some physical media such as diskette, tape, or CD. However, product 1 can be electronically delivered over communications lines to
20 the licensee and therefore might exist in the memory of the licensee's machine, rather than any physical media. In the case of a product such as music, radio programs and the like, product 1 may even be broadcast to the licensee's site for playback; thus, the product
25 1 would not even be "resident" in the licensee's machine.

The presently described embodiment allows the licensee to access the licensed product concurrent with the sending and receiving of datagram 3. In this
30 way, the present invention does not inconvenience the legitimate licensee; however, for sensitive licensed products such as confidential information, the license

check monitor 2 can prevent access to the product 1 until an authorization reply datagram 3 is received.

Further, monitor 2 could be realized as an integral part of product 1. Monitor 2 could also be implemented as: 1) a separate process which is the parent process of product 1 (Such a parent process would have the authority to cancel the use of product 1); 2) a single system level task which controls license checking of all products at the licensee's site; and 3) custom logic in a digital integrated circuit (the present invention could be implemented as hardware instead of software).

Also, though the above embodiment has been described as being implemented on a computer system network where operator messages are provided on a CRT monitor or the like, the invention may be practiced on other hardware platforms by incorporating appropriate changes known to those of ordinary skill in the art. For example, in an alternative hardware embodiment such as a music or video playback device, monitor 2 is invoked by the licensee's action of pushing the "play" or similar button, and in a broadcast music application or similar system, the monitor may be invoked simply by turning the device on. The processing of monitor 2 is as described in the presently described embodiment. However, when a denial message is received or generated, monitor 2 must be able to switch "play" to "off".

The presently described embodiment is designed to be used in conjunction with a connectionless UDP (User Datagram Protocol) in the TCP/IP protocol suite as an underlying protocol. However, the present invention could also be realized using a slower,

connectionless protocol such as electronic mail or a variety of connection protocols (e. g., File Transfer Protocols (FTP), Telnet).

5 It is noted that protocol suites quite different from TCP/IP could be used, such as ISO (International Standards Organization) protocol. In addition, datagrams 3 could be sent over telephone systems with communications protocols such as those specified by CCITT (Consultative Committee on International
10 Telephony and Telegraphy). In this case, telephone numbers could serve as network addresses 21, 22. Communications protocols for wireless communications such as cellular telephone can also be used to send the datagram 3.

15 Accordingly, all such modifications are intended to be included within the scope of this invention as defined by the following claims.

WHAT IS CLAIMED IS:

1. A method for monitoring the use of a licensed product, comprising the steps of:
 - generating, at regular time intervals,
5 datagrams including an address in a communications facility, said facility address identifying a licensee;
 - automatically sending said datagrams from at least one licensee's site over said facility to a
10 licensor's site while said licensed product is in use;
 - receiving said datagrams at said licensor's site;
 - storing an indication of receipt of each of said datagrams; and
 - 15 counting said datagrams from each licensee as an indication of the use by the licensee of said licensed product.
2. A method as in claim 1 further wherein:
 - said generating step includes the step of
20 incorporating a model number of said product in said datagrams; and
 - said counting step includes the step of separately counting datagrams for each product model number for each licensee.
- 25 3. A method as in claim 1, wherein said generating step includes the step of automatically obtaining said facility address that identifies said licensee from said facility without any data being provided by said licensee.

- 28 -

4. A method for controlling use of a licensed product comprising the steps of:

generating a request datagram including an address in a communications facility, said facility address identifying a licensee;

automatically sending said request datagram from at least one licensee's site over said facility to a licensor's site while said licensed product is in use;

receiving said request datagram at said licensor's site;

comparing said received request datagram with rules and license data at said licensor's site to determine if use of said licensed product is authorized;

sending a reply authorizing datagram to said licensee's site if use of said licensed product is approved; and

receiving said reply authorizing datagram at said licensee's site and denying the use of said product when no reply authorizing datagram is received.

5. A method as in claim 4, wherein:

said generating step includes the step of incorporating a model number of said product in said datagram;

said comparing step includes the step of comparing said rules and license data for a particular model number; and

said sending step includes the step of transmitting said reply datagram for each product model number.

- 29 -

6. A method as in claim 4, wherein said
generating step includes the step of automatically
obtaining said facility address that identifies said
licensee from said facility without any data being
5 provided by said licensee.

7. A method as in claim 4 further comprising
the step of sending a reply denial datagram if use of
said licensed product is not approved as determined in
said comparing step, said step of automatically
10 sending said request datagram from a licensee's site
including the step of resending said request datagram
if neither a reply authorizing datagram nor a reply
denial datagram is received from said licensor's site
within a predetermined time from sending said request
15 datagram from said licensee's site.

8. A method as in claim 4, wherein said step of
automatically sending said request datagram from said
licensee's site includes the step of sending a request
datagram at regular time intervals.

20 9. A method as in claim 4, wherein:
said generating step includes the step of
providing a datagram identification code within said
datagram;
said reply datagram sending step includes
25 the step of inserting the same datagram identification
code in said reply datagram; and
said reply receiving step rejects said reply
authorizing datagram if the datagram identification
code included in said reply authorizing datagram does

SUBSTITUTE SHEET

- 30 -

not match the datagram identification code included in said request datagram.

10. A method as in claim 4, wherein:

5 said comparing step includes the step of comparing said facility address that identifies said licensee with a list of valid licensee addresses to determine if said facility address is a valid address; and

10 said reply authorizing datagram is not sent if said facility address that identifies said licensee is not valid.

11. A method as in claim 10 further comprising the step of sending a reply denial datagram if said facility address that identifies said licensee is not
15 valid.

12. A method as in claim 4, wherein:

said comparing step includes the step of comparing a license expiration date with a date at which said datagram is received; and

20 said reply authorizing datagram is not sent if the license expiration date is later than the date at which said datagram is received.

13. A method as in claim 12, further comprising the step of sending a reply denial datagram if the
25 license expiration date is later than the date at which said datagram is received.

14. A method as in claim 4, wherein:

SUBSTITUTE SHEET

said comparing step includes the step of checking currentness of payments from said license; and

5 said reply authorizing datagram is not sent if payment is overdue.

15. A method as in claim 14, further comprising the step of sending a reply denial datagram if payment is overdue.

16. A method as in claim 4, wherein:

10 said generating step includes the step of incorporating in said datagram data indicative of the number of processes currently using said product at said licensee's site;

15 said comparing step includes the step of comparing the number of processes using said product at the licensee's site to an authorized number; and

said reply authorizing datagram is not sent if said number of processes using said product exceeds said authorized number.

20 17. A method as in claim 16, further comprising the step of sending a reply denial datagram if said number of processes using said product exceeds said authorized number.

25 18. A method as in claim 4, wherein said sending step includes the steps of sending said reply authorizing datagram when use of said product is approved and sending a reply denial datagram when use of said product is not approved, said receiving step

denying use of said product when said reply denial datagram is received.

19. A method as in claim 18, wherein said receiving and denying step denies use of said product
5 when neither a reply authorizing datagram nor a reply denial datagram is received within a predetermined time after said request datagram is sent.

20. A method as in claim 18, further comprising the step of indicating, at a licensee's site, a reason
10 for denial when said reply denial datagram is received.

21. A method as in claim 4, wherein:
said licensed product comprises an executable portion and a data portion; and
15 said method further comprises a step of controlling use of said data portion with said executable portion.

22. A method as in claim 4 further comprising a step of allowing use of said licensed product before
20 a reply datagram is received.

23. A system for controlling licensed product comprising:
a communications facility to which at least one licensee having a license for operating a licensed
25 product from the licensor is connected;
monitoring means, connected to said facility at a site of each said licensee, for generating a request datagram including an address of said licensee

on said facility and transmitting said request datagram over said facility to a site of said licensor, and for receiving and processing a reply datagram; and

5 controlling means, connected to said facility at said licensor's site, for receiving said request datagram, comparing said request datagram with rules and license data to determine if use of said licensed product is authorized and sending a reply
10 authorizing datagram to said licensee's site if use of said product is approved; and

 said monitoring means including means for denying use of said licensed product when no reply authorizing datagram is received.

15 24. A system as in claim 23, wherein:

 said monitoring means sends request datagrams at regular time intervals during use of said licensed product; and

20 said controlling means further comprises means for counting said request datagrams received at said controlling means and means for computing an amount to be billed to said licensee in response to said counting.

 25. A system as in claim 23 wherein:

25 said monitoring means incorporates a model number for said product in said request datagram; and

 said controlling means comprises means for counting datagrams for each product model number for each licensee, in order to compute an amount to be
30 billed to each licensee.

26. A system as in claim 23, wherein said monitoring means automatically obtains said facility address of said licensee from said facility without any input from said licensee.

5 27. A system as in claim 23, wherein:
 said controlling means sends a reply denial datagram to said licensee's site if use of said product is not approved; and
 said monitoring means resends said request
10 datagram if no reply authorizing datagram and no reply denial datagram is received within a predetermined period of time after said requesting datagram is sent.

 28. A system as in claim 23, wherein said
15 monitoring means transmits request datagrams at predetermined time intervals.

 29. A system as in claim 23, wherein:
 said monitoring means incorporates a unique
 identification code in said request datagram;
 said controlling means incorporates the same
20 request datagram identification code in said reply authorizing datagram; and
 said monitoring means rejects any reply
 authorizing datagram which does not include the same
 identification code as included in said request
25 datagram.

 30. A system as in claim 23, wherein said
controlling means compares said facility address of
said licensee with a list of valid licensee facility
addresses and does not generate a reply authorizing

datagram if said facility address of said licensee is not valid.

31. A system as in claim 30, wherein said controlling means sends a reply denial datagram when
5 said facility address is not valid.

32. A system as in claim 23, wherein said controlling means compares an expiration date of a license of said product with a date at which said request datagram is received by said controlling
10 means, and does not generate a reply authorizing datagram, thus denying use of said product, if the license expiration date is earlier than the date at which said request datagram is received.

33. A system as in claim 32, wherein said
15 controlling means sends a reply denial datagram if the license expiration date is earlier than the date at which said request datagram is received.

34. A system as in claim 23, wherein said controlling means generates a reply authorizing
20 datagram, thus denying use of said product, if a payment for the use of said product is overdue.

35. A system as in claim 34, wherein said controlling means sends a reply denial datagram if payment for the use of said product is overdue.

25 36. A system as in claim 23, wherein:
said monitoring means includes in said request datagram data indicative of the number of

processes, at a licensee's site, currently using said product; and

5 said controlling means does not generate a reply authorizing datagram, thus denying a use of said product, if more than a predetermined number of processes using said product are running at the licensee's site.

10 37. A system as in claim 36, wherein said controlling means sends a reply denial datagram if more than said predetermined number of processes using said product are running at the licensee's site.

 38. A system as in claim 23, wherein said controlling means sends a reply denial datagram if use of said product is not approved.

15 39. A system as in claim 38, wherein said monitoring means denies use of said licensed product when no reply authorizing datagram and no reply denial datagram is received within a predetermined time from the sending of said request datagram.

20 40. A system as in claim 38, further comprising means for indicating, at a licensee's site, a reason for denial when said reply denial datagram is received.

25 41. A system as in claim 23, wherein:
 said licensed product comprises an executable portion and a data portion; and

said system further comprises means for controlling use of said data portion with said executable portion.

5 42. A system as in claim 41, wherein said data portion controlling means is disposed within said executable portion.

10 43. A system as in claim 41, wherein said data portion controlling means comprises a first partial controlling means disposed within said executable portion and a second partial controlling means disposed within said monitoring means.

15 44. A system as in claim 23, wherein said monitoring means includes means for permitting use of said licensed product before a reply datagram is received.

20 45. A system for monitoring product comprising:
a communications facility to which at least one licensee having a license for operating a licensed product from a licensor is connected;
monitoring means, connected to said facility at a site of each said licensee, for generating datagrams including an address of said licensee on said facility and transmitting said datagrams at periodic intervals over said facility to a site of said licensor; and

25 control means, connected to said facility at said licensor's site, for receiving said request datagrams, storing an indication of receipt of each of said datagrams and counting said datagrams from each

licensee as an indication of the use by the licensee of said licensed product.

46. A system as in claim 45, wherein said monitoring means automatically obtains said facility address of said licensee from said facility without
5 any input from said licensee.

47. A system as in claim 45, wherein:
said monitoring means incorporates a product model number in said request datagrams; and
10 said controlling means separately counts request datagrams for each product model number for each licensee.

48. A method for monitoring the use of a licensed product comprising the steps of:
15 generating, at regular time intervals, datagrams including an address in a communications facility, said facility address identifying a licensee; and
automatically sending said datagrams from at
20 least one licensee's site over said communications facility to a licensor's site while said licensed product is in use.

49. A method as in claim 48 further wherein:
said generating step includes the step of
25 incorporating a model number of said product in said datagrams.

50. A method as in claim 48, wherein said generating step includes the step of automatically

obtaining said facility address that identifies said licensee from said communications facility without any data being provided by said licensee.

5 51. A method for controlling use of a licensed product comprising the steps of:

generating a request datagram including a facility address that identifies a licensee in a communications facility;

10 automatically sending said request datagram from a licensee's site over said communications facility to a licensor's site while said licensed product is in use; and

15 receiving a reply authorizing datagram at said licensee's site and denying the use of said product when no reply authorizing datagram is received.

52. A method as in claim 51 wherein:

20 said generating step includes the step of incorporating a model number of said product in said datagram.

53. A method as in claim 51, wherein said generating step includes the step of automatically obtaining said facility address that identifies said licensee from said communications facility without any
25 data being provided by said licensee.

54. A method as in claim 51, wherein:

said reply datagram is one of at least a reply authorization datagram and a reply denial datagram; and

said step of automatically sending said request datagram from a licensee's site includes a step of resending said request datagram if neither a reply authorizing datagram nor a reply denial datagram is received within a predetermined time from sending said request datagram from said licensee's site.

55. A method as in claim 51, wherein said step of automatically sending said request datagram from said licensee's site includes the step of sending a request datagram at regular time intervals.

56. A method as in claim 51, wherein:
said generating step includes the step of providing a datagram identification code within said datagram; and
said reply receiving step rejects said reply authorizing datagram if the datagram identification code included in said reply authorizing datagram does not match the datagram identification code included in said request datagram.

57. A method as in claim 51, wherein:
said generating step includes the step of incorporating in said datagram data indicative of the number of processes currently using said product at said licensee's site.

58. A method as in claim 51, further comprising the steps of:
receiving a reply denial datagram; and
displaying, at a licensee's site, a reason for denial when said reply denial datagram is received.

59. A method as in claim 51, wherein:

said licensed product comprises an executable portion and a data portion; and

5 said method further comprises a step of controlling use of said data portion with said executable portion.

60. A method as in claim 51 further comprising a step of allowing use of said licensed product before a reply datagram is received.

10 61. A system for controlling a licensed product comprising:

a communications facility to which at least one licensee is connected;

15 monitoring means, connected to said communications facility at a site of each said licensee, for generating a request datagram including an address of said licensee on said communications facility and transmitting said request datagram over said communications facility, and for receiving and
20 processing a reply authorizing datagram; and

means for denying use of said product when no reply authorizing datagram is received.

62. A system as in claim 61, wherein:

25 said monitoring means sends request datagrams at regular time intervals during use of said licensed product.

63. A system as in claim 61 wherein:

said monitoring means incorporates a model number for said product in said request datagram.

64. A system as in claim 61, wherein said monitoring means automatically obtains said facility address of said licensee from said communications facility without any input from said licensee.

65. A system as in claim 61, wherein:
said monitoring means resends said request datagram if no reply authorizing datagram and no reply denial datagram is received within a predetermined period of time after said requesting datagram is sent.

66. A system as in claim 61, wherein said monitoring means transmits request datagrams at predetermined time intervals.

67. A system as in claim 61, wherein:
said monitoring means incorporates a unique identification code in said request datagram; and
said monitoring means rejects any reply authorizing datagram which does not include the same identification code as included in said request datagram.

68. A system as in claim 61, wherein:
said monitoring means includes in said request datagram data indicative of the number of processes, at a licensee's site, currently using said product.

69. A system as in claim 61, wherein:

5 said monitoring means denies use of said licensed product when no reply authorizing datagram and no reply denial datagram is received within a predetermined time from the sending of said request datagram.

70. A system as in claim 61, further comprising means for indicating, at a licensee's site, a reason for denial when a reply denial datagram is received.

10 71. A system as in claim 61, wherein:
said licensed product comprises an executable portion and a data portion; and
said system further comprises means for controlling use of said data portion with said executable portion.

15 72. A system as in claim 71, wherein said data portion controlling means is disposed within said executable portion.

20 73. A system as in claim 71, wherein said data portion controlling means comprises a first partial controlling means disposed within said executable portion and a second partial controlling means disposed within said monitoring means.

25 74. A system as in claim 61, wherein said monitoring means includes means for permitting use of said licensed product before a reply datagram is received.

75. A system for monitoring a licensed product comprising:

a communications facility to which at least one licensee is connected;

5 monitoring means, connected to said communications facility at a site of each said licensee, for generating datagrams including an address of said licensee on said communications facility and transmitting said datagrams at periodic
10 intervals over said communications facility.

76. A system as in claim 75, wherein said monitoring means automatically obtains said communications facility address of said licensee from said communications facility without any input from
15 said licensee.

77. A system as in claim 75, wherein:
said monitoring means incorporates a product model number in said request datagrams.

78. A method for monitoring the use of a
20 licensed product comprising the steps of:

receiving datagrams at a licensor's site on a communications facility having at least one licensee's site thereon, said datagrams being generated at regular time intervals and including a
25 facility address that identifies a licensee in said communications facility;

storing an indication of receipt of each of said datagrams; and

30 counting said datagrams as an indication of the use of said licensed product.

79. A method as in claim 78 further wherein:
said datagrams include a model number of
each product; and

5 said counting step includes the step of
separately counting datagrams for each product model
number for each licensee.

80. A method for controlling use of a licensed
product comprising the steps of:

10 receiving a request datagram at a licensor's
site on a communications facility having at least one
licensee's site thereon, said request datagram
including a facility address identifying a licensee
and being automatically sent over said communications
15 facility to said licensor's site while said licensed
product is in use;

comparing said received request datagram
with rules and license data at said licensor's site to
determine if use of said licensed product is
authorized; and

20 sending a reply authorizing datagram if use
of said licensed product is approved.

81. A method as in claim 80 wherein:

said datagrams include a model number of
said product;

25 said comparing step includes the step of
comparing said rules and license data for a particular
model number; and

said sending step includes the step of
transmitting said reply datagram for each product
30 model number.

82. A method as in claim 80 further comprising the step of sending a reply denial datagram if use of said licensed product is not approved as determined in said comparing step.

5 83. A method as in claim 80, wherein:
 said datagrams include a datagram
 identification code; and
 said reply datagram sending step includes
 the step of inserting the same datagram identification
10 code in said reply datagram.

 84. A method as in claim 80, wherein:
 said comparing step includes the step of
 comparing said facility address that identifies said
 licensee with a list of valid licensee addresses to
15 determine if said facility address is a valid address;
 and
 said reply authorizing datagram is not sent
 if said facility address that identifies said licensee
 is not valid.

20 85. A method as in claim 84 further comprising
 the step of sending a reply denial datagram if said
 facility address that identifies said licensee is not
 valid.

 86. A method as in claim 80, wherein:
25 said comparing step includes the step of
 comparing a license expiration date with a date at
 which said datagram is received; and

- 47 -

said reply authorizing datagram is not sent if the license expiration date is later than the date at which said datagram is received.

5 87. A method as in claim 86, further comprising the step of sending a reply denial datagram if the license expiration date is later than the date at which said datagram is received.

10 88. A method as in claim 80, wherein:
said comparing step includes the step of checking currentness of payments from said license;
and
said reply authorizing datagram is not sent if payment is overdue.

15 89. A method as in claim 88, further comprising the step of sending a reply denial datagram if payment is overdue.

20 90. A method as in claim 80, wherein:
said datagrams include data indicative of the number of processes currently using said product at said licensee's site;
said comparing step includes the step of comparing a number of processes using said product to an authorized number; and
25 said reply authorizing datagram is not sent if said number of processes using said product exceeds said authorized number.

91. A method as in claim 90, further comprising the step of sending a reply denial datagram if said

number of processes using said product exceeds said authorized number.

92. A method as in claim 80, wherein said sending step includes the steps of sending said reply
5 authorizing datagram when use of said product is approved and sending a reply denial datagram when use of said product is not approved.

93. A system for controlling a licensed product comprising:

10 a communications facility to which at least one licensee and a licensor are connected at a licensee's site and at a licensor's site, respectively; and

15 controlling means, connected to said communications facility at said licensor's site, for: receiving a request datagram, said request datagram including an address of said licensee on said communications facility and being transmitted over
20 said communications facility to a site of said licensor; comparing said request datagram with rules and license data to determine if use of said licensed product is authorized; and sending a reply authorizing datagram to said licensee's site if use of said product is approved.

25 94. A system as in claim 93, wherein:

said request datagrams are sent at regular time intervals during use of said licensed product; and

30 said controlling means comprises means for counting said request datagrams received at said

controlling means and means for computing an amount to be billed to said licensee in response to said counting.

95. A system as in claim 93 wherein:

5 said datagrams include a model number for said product; and

 said controlling means comprises means for counting datagrams for each product model number for each licensee, in order to compute an amount to be
10 billed to each licensee.

96. A system as in claim 93, wherein:

 said controlling means sends a reply denial datagram to said licensee's site if use of said product is not approved.

15 97. A system as in claim 93, wherein:

 said datagrams include a unique identification code; and

 said controlling means incorporates the same request datagram identification code in said reply
20 authorizing datagram.

98. A system as in claim 93, wherein said controlling means compares said facility address of said licensee with a list of valid licensee facility addresses and does not generate a reply authorizing
25 datagram if said facility address of said licensee is not valid.

99. A system as in claim 98, wherein said controlling means sends a reply denial datagram when said facility address is not valid.

5 100. A system as in claim 93, wherein said controlling means compares an expiration date of a license of said product with a date at which said request datagram is received by said controlling means, and does not generate a reply authorizing datagram, thus denying use of said product, if the
10 license expiration date is earlier than the date at which said request datagram is received.

15 101. A system as in claim 100, wherein said controlling means sends a reply denial datagram if the license expiration date is earlier than the date at which said request datagram is received.

102. A system as in claim 93, wherein said controlling means generate a reply authorizing datagram, thus denying use of said product, if a payment for the use of said product is overdue.

20 103. A system as in claim 102, wherein said controlling means sends a reply denial datagram if payment for the use of said product is overdue.

104. A system as in claim 93, wherein:
said datagrams include data indicative of
25 the number of processes, at a licensee's site, currently using said product; and
said controlling means does not generate a reply authorizing datagram, thus denying a use of said

product, if more than a predetermined number of processes using said product are running at the licensee's site.

5 105. A system as in claim 104, wherein said controlling means sends a reply denial datagram if more than said predetermined number of processes using said product are running at the licensee's site.

10 106. A system as in claim 93, wherein said controlling means sends a reply denial datagram if use of said product is not approved.

107. A system as in claim 93, wherein:
said licensed product comprises an executable portion and a data portion; and
said system further comprises means for
15 controlling use of said data portion with said executable portion.

108. A system as in claim 107, wherein said data portion controlling means is disposed within said executable portion.

20 109. A system for monitoring a licensed product comprising:

a communications facility to which at least one licensee and a licensor are connected at a licensee's site and at a licensor's site,
25 respectively; and

control means, connected to said communications facility at a licensor's site, for: receiving request datagrams, said request datagrams

including an address of said licensee on said communications facility and being transmitted at periodic intervals over said communications facility to said licensor's site; storing an indication of receipt of each of said datagrams; and counting said datagrams from each licensee as an indication of the use by the licensee of said licensed product.

110. A system as in claim 110, wherein:
said request datagrams include a product model number; and
said controlling means separately counts request datagrams for each product model number for each licensee.

FIG. 1

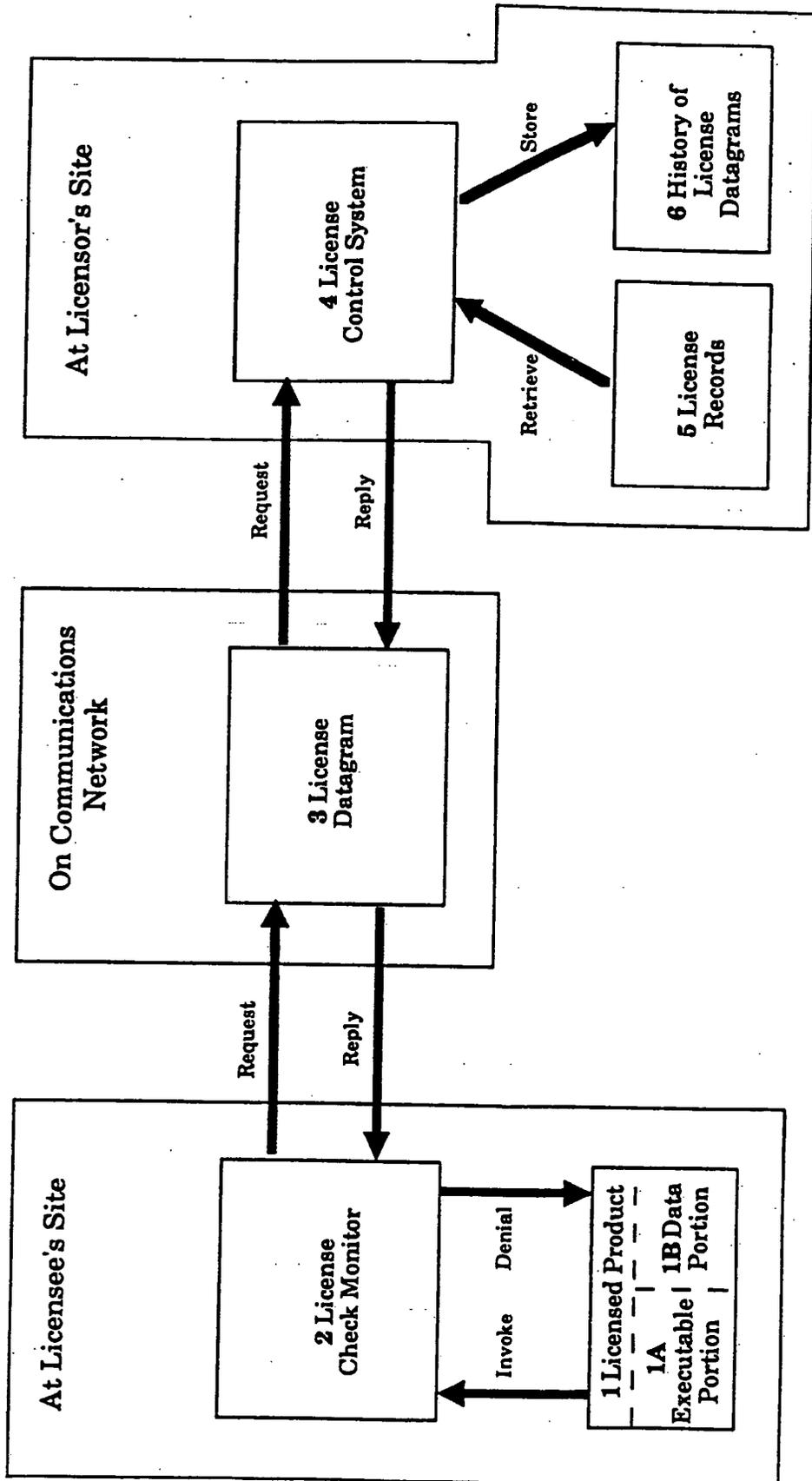


FIG. 2

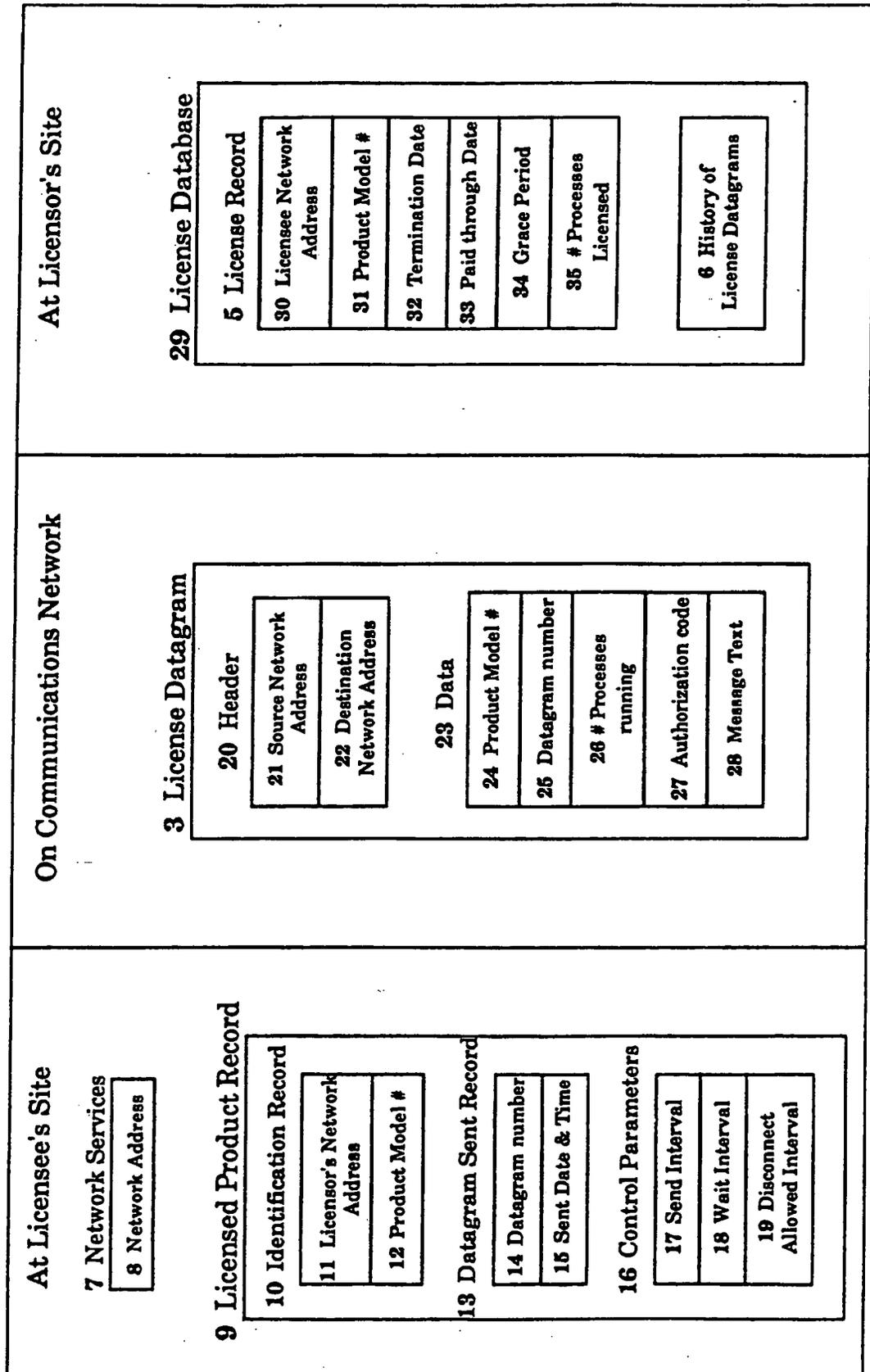


FIG. 3

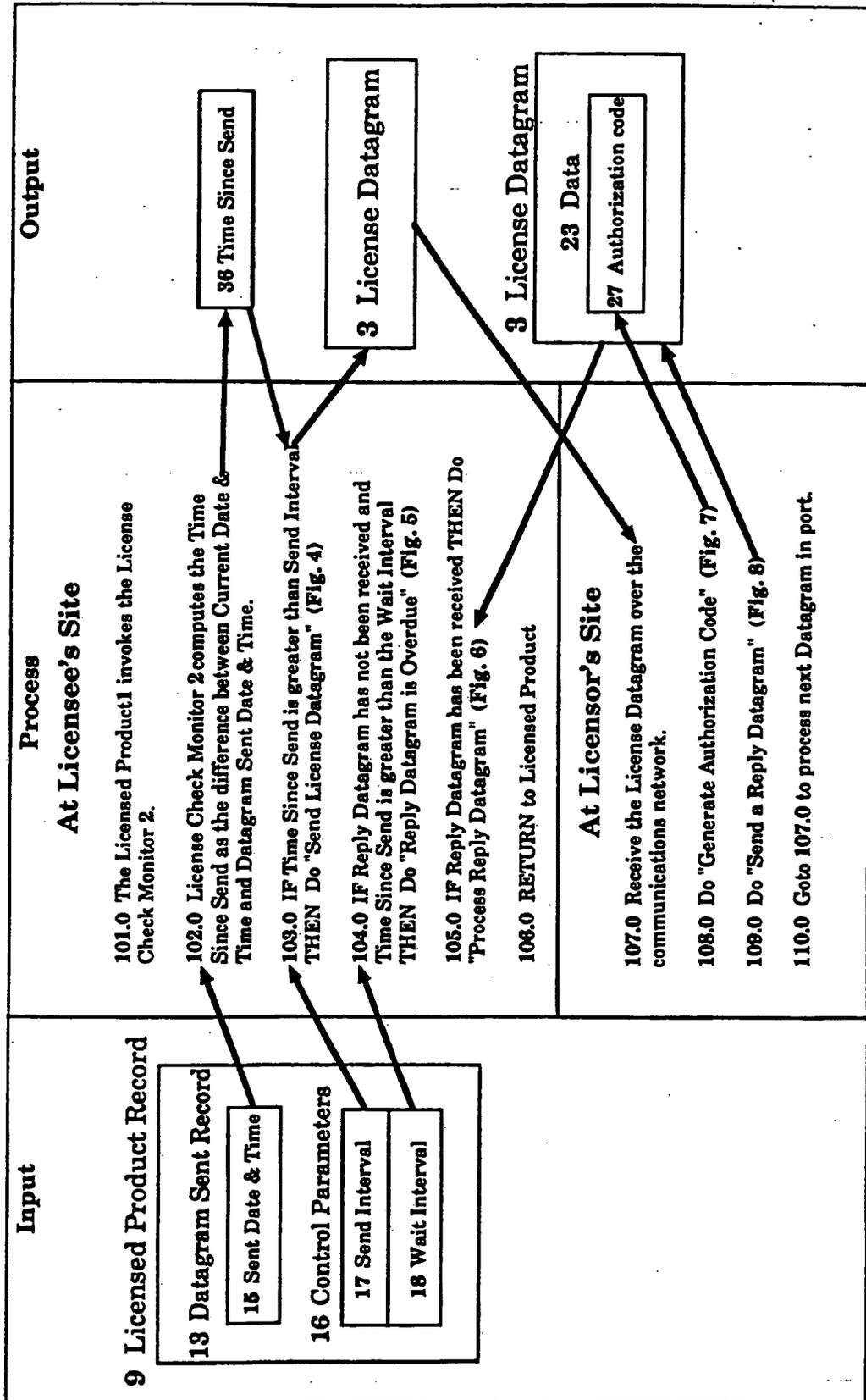


FIG. 4

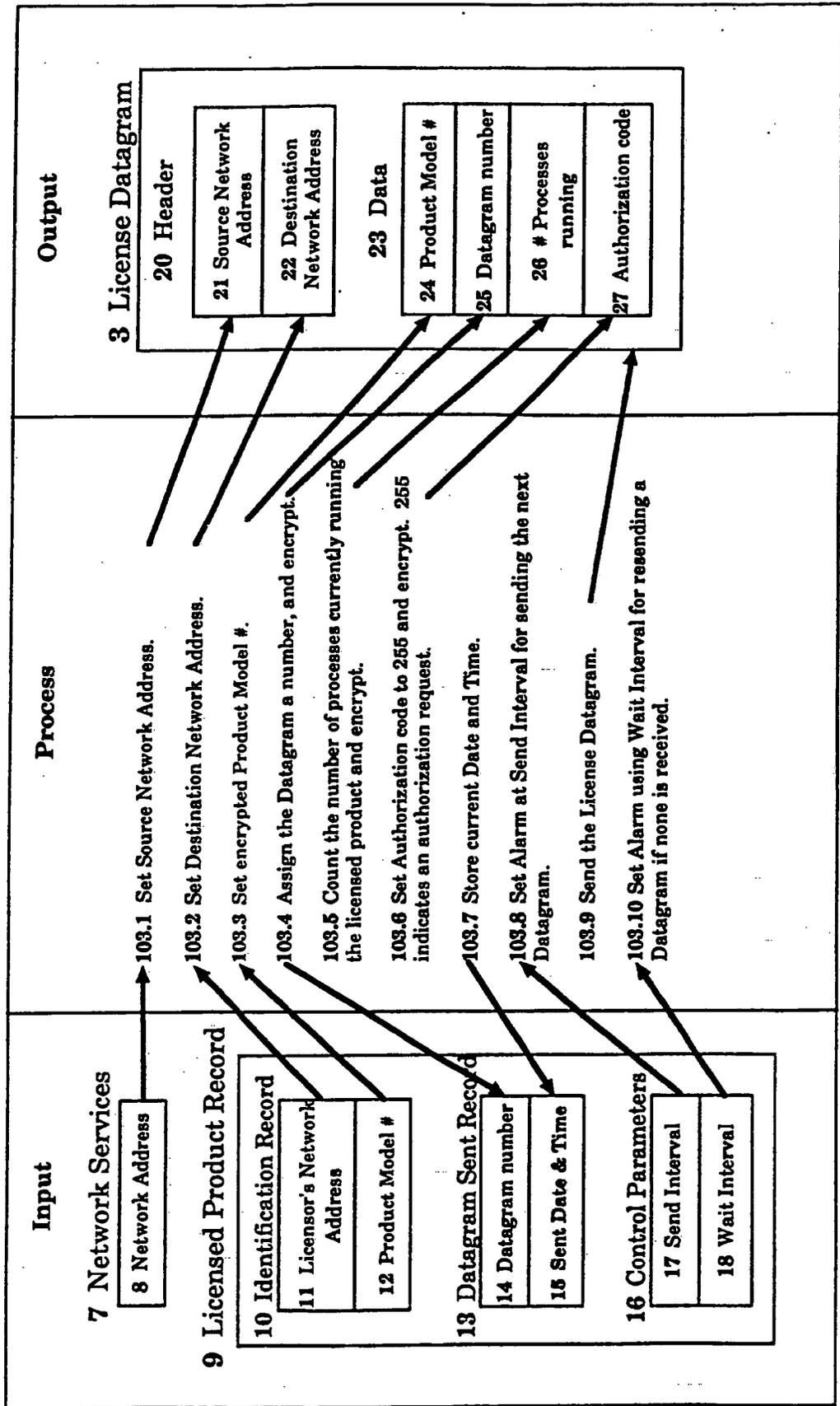


FIG. 5

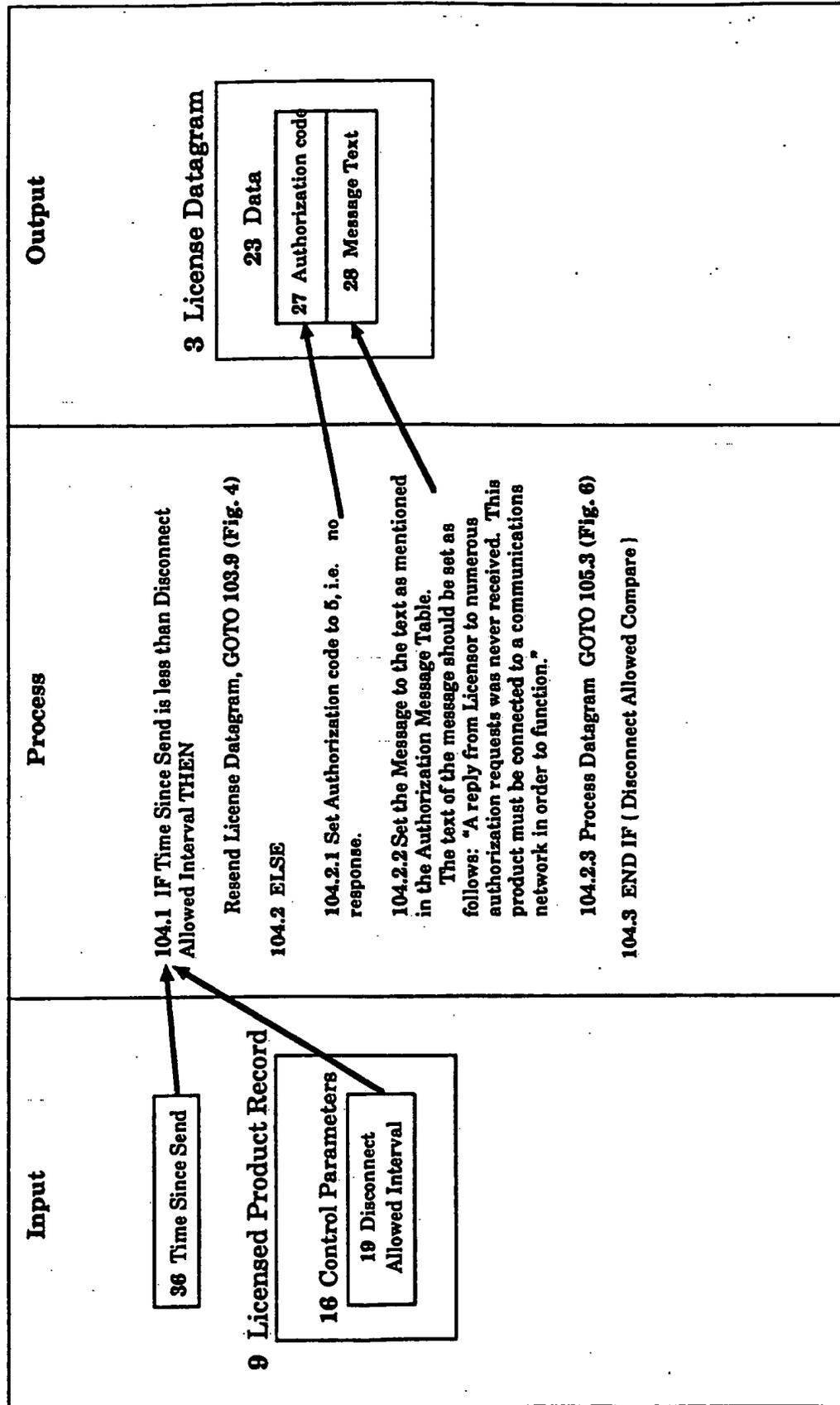


FIG. 6

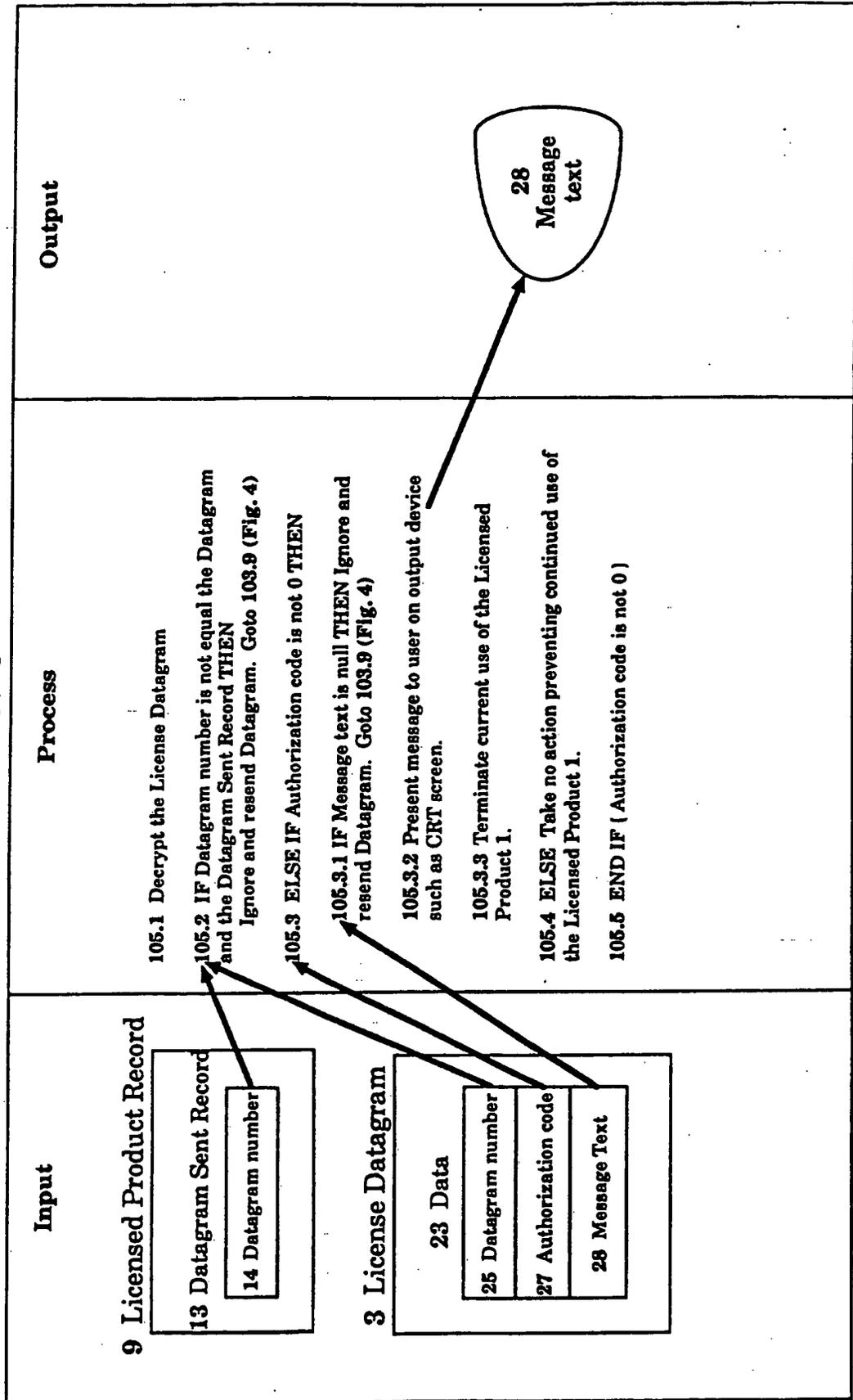


FIG. 7

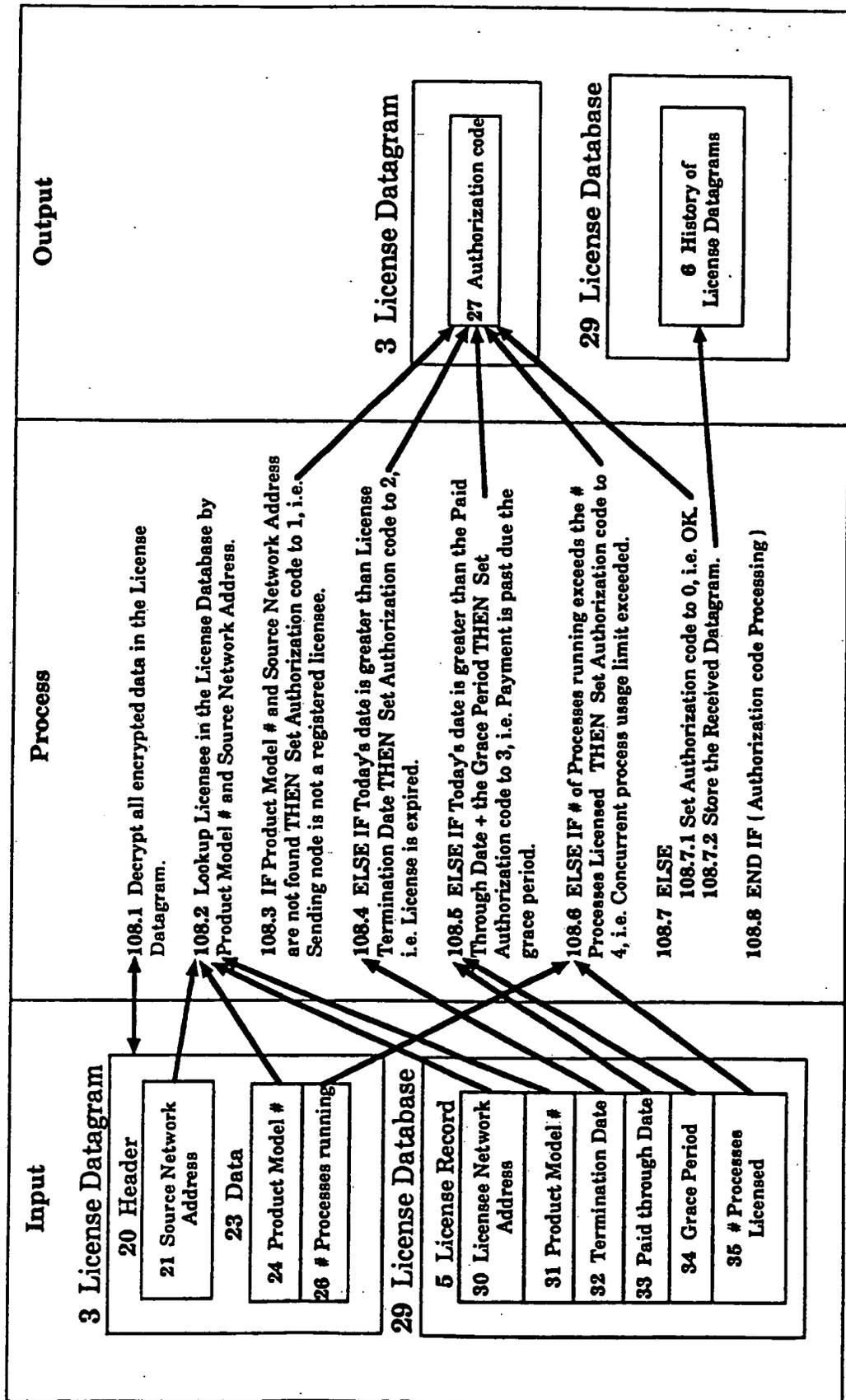
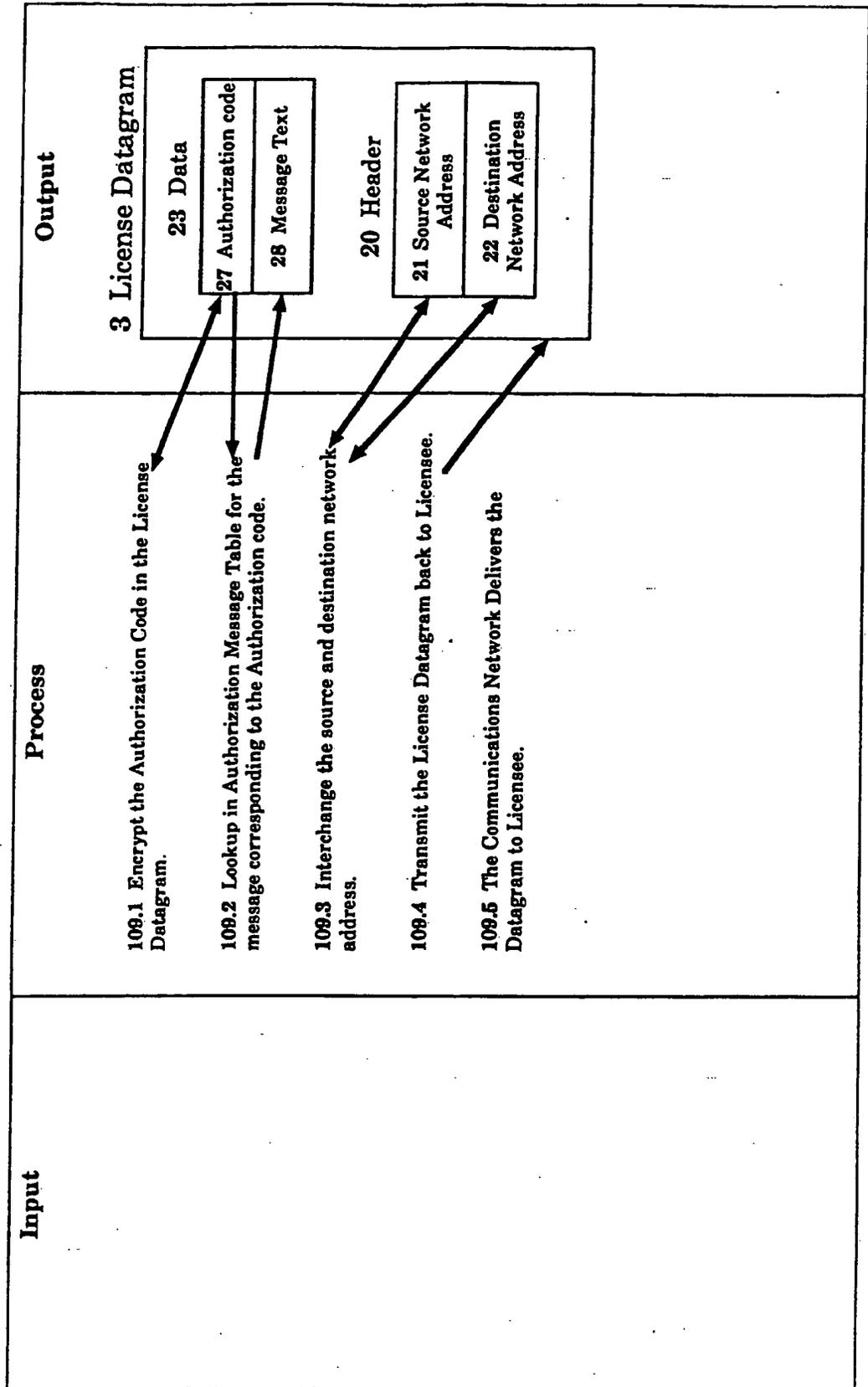


FIG. 8



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US92/05387

A. CLASSIFICATION OF SUBJECT MATTER		
IPC(5) :G06F 11/34; H04L 9/00 US CL :395/725; 380/4 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) U.S. : 364/406; 380/25		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) APS DATABASE: Software#, information, usage, monitor?, Licens?		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,P	US,A, 5,103,476 (WAITE ET AL) 07 APRIL 1992 See entire text.	1-110
Y,P	US,A, 5,050,213 (SHEAR) 17 SEPTEMBER 1991 See column 6, lines 27-51.	1-6,9-21,23-26,29-43,45-53,56-59,61-64,67-73,75-110
Y,P	US,A, 5,047,928 (WIEDEMER) 10 SEPTEMBER 1991 See col. 6, lines 16-54.	1-6,9-21,23-36,29-43,45-53,56-59,61-64,67-73,75-110
Y	US,A, 5,023,907 (JOHNSON ET AL) 11 JUNE 1991 See entire document.	1-110
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
<ul style="list-style-type: none"> * Special categories of cited documents: *A* document defining the general state of the art which is not considered to be part of particular relevance *E* earlier document published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed 	<ul style="list-style-type: none"> *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art *&* document member of the same patent family 	
Date of the actual completion of the international search 05 AUGUST 1992		Date of mailing of the international search report 04 NOV 1992
Name and mailing address of the ISA/ Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. NOT APPLICABLE		Authorized officer <i>Kenneth S. Kim</i> KENNETH S. KIM Telephone No. (703) 308-1634

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US92/05387

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US,A, 5,014,234 (EDWARDS, JR.) 07 MAY 1991 See col. 3, lines 4-16.	1-110
Y	US,A, 5,010,571 (KATZNELSON) 23 APRIL 1991 See entire document.	1-6,9-21,23-26,29- 43,45-53,56-59,61- 64,67-73,75-110.
Y	MACMILLAN Publishing Company, 1985, WILLIAM STALINGS, Data and Computer Communications. p199-203.	1-110
Y,P	US,A, 5,113,519 (JOHNSON ET AL) 12 MAY 1992 See col. 6, lines 36-68.	1-110
Y	US,A, 4,937,863 (ROBERT ET AL) 26 JUNE 1990 See col. 3, lines 25-40.	1-6,9-21,23-26,29- 43,45-53,56-59,61- 64,67-73,75-110

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*



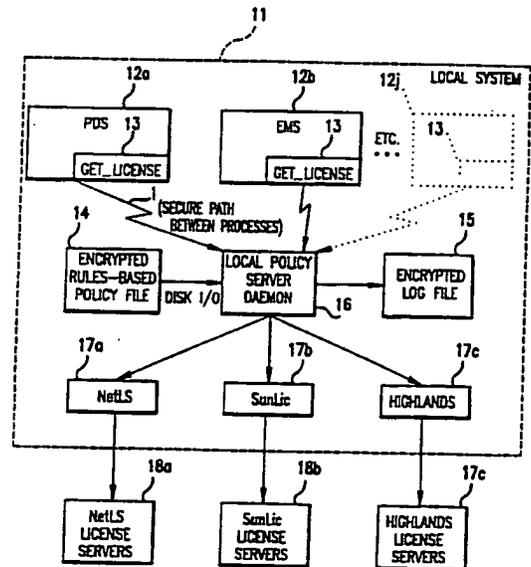
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁵ : G06F 1/00, 11/34</p>	<p>A1</p>	<p>(11) International Publication Number: WO 93/11480 (43) International Publication Date: 10 June 1993 (10.06.93)</p>
<p>(21) International Application Number: PCT/US92/10215 (22) International Filing Date: 24 November 1992 (24.11.92) (30) Priority data: 07/798,934 27 November 1991 (27.11.91) US (71) Applicant: INTERGRAPH CORPORATION [US/US]; One Madison Industrial Park, Huntsville, AL 35894 (US). (72) Inventors: BAINS, Jeffrey, E. ; 134 Michli Road, Madison, AL 35758 (US). CASE, Willard, W. ; 104 Arden Avenue, Madison, AL 35758 (US). (74) Agents: SUNSTEIN, Bruce, D. et al.; Bromberg & Sunstein, 10 West Street, Boston, MA 02111 (US).</p>		<p>(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i></p>

(54) Title: **SYSTEM AND METHOD FOR NETWORK LICENSE ADMINISTRATION**

(57) Abstract

Disclosed is a system for administration, on a computer network, of license terms (a so-called license server) for a software product (12a, 12b... 12j) provided to said network. Said license server (17c, 18a, 18b) being realized by one of the network computers and which tasks comprise e.g. tracking of a software product (12a, 12b... 12j) usage in the system, issuing usage permits (licenses) to the different network users in accordance with predefined conditions, monitoring expirations and violations (e.g. the maximum number of users simultaneously using a software product) of issued licenses and when necessary, withdrawing issued software product licenses. In one embodiment, the system includes a policy server database (14) maintained on each node (11) of the system, where said predefined conditions are specified under which usage of a software product (12a, 12b... 12j) is permitted at the respective system nodes (11). Each node also has a policy server "daemon" (16) in association with said policy server database (14) for interaction with the license server (17c, 18a, 18b) in order to enforce license terms for a software product.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NL	Netherlands
BE	Belgium	GN	Guinea	NO	Norway
BF	Burkina Faso	GR	Greece	NZ	New Zealand
BG	Bulgaria	HU	Hungary	PL	Poland
BJ	Benin	IE	Ireland	PT	Portugal
BR	Brazil	IT	Italy	RO	Romania
CA	Canada	JP	Japan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SK	Slovak Republic
CI	Côte d'Ivoire	LJ	Liechtenstein	SN	Senegal
CM	Cameroon	LK	Sri Lanka	SU	Soviet Union
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	MC	Monaco	TG	Togo
DE	Germany	MG	Madagascar	UA	Ukraine
DK	Denmark	ML	Mali	US	United States of America
ES	Spain	MN	Mongolia	VN	Viet Nam
FI	Finland				

- 2 -

software that is licensed for concurrent or simultaneous use. Some licensors use hardware locks that attach to a parallel printer port or a serial port on a machine; each time the software is activated, it looks for a specified code, in the hardware lock, as a condition for operation of the software. Using hardware locks resolves the problem of unauthorized moving of software among machines; however, hardware locks do not handle multiple software products on a single machine, and they require time and expense to deliver to the end user.

When computer software products are used in a network environment (which may include computers running in various roles as workstations and servers of various types linked together over a data path), additional licensing challenges are present. For example, a network may permit a user at one node (which may be a terminal or workstation, for instance) to utilize a software product running at another node (which may be the network server or even another workstation). Consequently, the terms of the single-computer type of software license might not cover the usage of the software product on the network, or worse still (from the point of view of the licensor) might actually permit such a usage without additional compensation to the licensor. One approach to network licensing is to grant permission to use the program based on all of the nodes on the network, and to require a license for each node. Then typically the license fee may be increased as the number of nodes on the network increases. Another approach bases the license fee for a software product running on a network on the total number of individual users who might actually run the software, regardless of the number of nodes either on the network or running the software product at a given time. These approaches, however, have usually required the cooperation of the licensee, because additional nodes may be added to the network, or additional users may utilize the software, without the knowledge of the licensor, who is typically not present on the premises of the licensee. The licensor may

- 3 -

reserve the right to audit the licensee's site, but such an audit is intrusive, expensive, and may alienate potential or actual customers for licenses. Although other approaches exist under which one might charge a single fee per server
5 or per site or per entity, often on an individually negotiated basis, these approaches are often impractical or inflexible, in that they also typically do not take into account the possible wide variation over time in the number of nodes or users and also require reliance on licensee
10 cooperation.

The same circumstances that make license enforcement difficult for the licensors of software products for a network environment also make license compliance difficult for the conscientious administrator, for example, of a
15 Management Information System (MIS) or Computer Aided Design (CAD) department of a company using software products. The administrator may be called upon to ensure that the number of workstations using a variety of software products in a network environment complies with the terms of a variety of
20 license agreements. Such an administrator may have to develop and promulgate a series of directives about the terms of permitted workstation usage and must depend primarily upon the goodwill and voluntary compliance of unit personnel with such directives.

25 Recently it has become practical in some network environments to determine and limit the number of nodes that may access a software product at a given time, and to charge a license fee based on the maximum number of nodes that are permitted to use the software product concurrently. This is
30 called "concurrent licensing". In these environments, a computer program, acting as "librarian" and running on a computer node designated as a license server, is typically used to distribute license keys (sometimes called "tokens") over the network to nodes requesting access to run a
35 software product; the number of keys is tracked by the librarian; and if at a given time, the permitted maximum number of keys would be exceeded by usage of the software

product on a requesting node, the node can be denied, at such time, access to invoke the software product.

Examples of software-based concurrent licensing arrangements may be found in Unix applications running in connection with software products sold under the trademarks NetLS (available from Gradient Technologies, Inc., 577 Main Street, Suite 4, Hudson, Massachusetts 01749), and SunLic (available from Sun Microsystems, Inc., Mountain View, California), and Flexible License Manager (available from Highland Software, Inc., 1001 Elwell Court, Palo Alto, California 94303). However these arrangements suffer from a number of disadvantages. NetLS, for example, includes mechanisms for tracking which nodes have been given keys to run a given software product and the number of keys available for running such software product. However, it is up to the designers of each software product to program such product to implement the terms of any license agreement, and, in particular, to program into the product calls to the NetLS software to provide information to the computer running the software product and to write code in the applicable product to prevent use of the product when the license terms have not been obeyed. Thus a computer system utilizing ten different software products that rely on NetLS for license enforcement will generally have ten different substantial software portions (one in each computer product) to achieve license enforcement. In addition to this complexity, if the license server running NetLS fails, or if the network itself fails, then a workstation loaded with the software product cannot run the software product, since the product requires NetLS interaction to be activated.

The foregoing difficulties are applicable generally not just to NetLS but to "metering software" generally. The Microcomputer Managers Association has issued a White Paper (October 2, 1991), reprinted in Infoworld, pages 46-42 (October 14, 1991) on the problems of network licensing, Commenting on the problem that each software product requires its own interface to the metering software (as well

- 5 -

as possible input of administrative information), the White Paper suggests that "[i]t makes much more sense to have a single package provide the metering for all application software on the network." Infoworld (October 14, 1991),
5 supra, at page 51, column 4. Such an approach has its own difficulty, however. Each application would still have to interface with the single metering package, and the interface to such a package must somehow deal with the varying licensing terms of each software product. Moreover,
10 with the metering package running on the license server, a failure of the server or the network would prevent all software applications from running anywhere on the network.

Summary of the Invention

In a preferred embodiment, the present invention
15 provides an improved system for administration, on a computer network, of license terms for a software product on the network. The improved system is of the type having an arrangement, such as NetLS, for tracking software product usage, associated with one of the computers acting as a
20 license server. This arrangement permits the license server (i) to identify the current set of nodes that are using the software product at a given time, (ii) to handle license data concerning conditions under which usage of the software product is permitted at any given node, and (iii) to
25 determine whether at any given time the conditions would still be satisfied if a given node is then added to this set of nodes. The software product may thus include instructions to interface with the license server to cause enforcement of the license terms. The improvement, in one
30 embodiment, to the system includes a policy server database maintained on each node, containing data specifying conditions under which usage of the software product is permitted on such node. Each node also has a policy server "daemon" (which may be implemented in software) in
35 association with the corresponding policy server database, for (i) communicating with the license server, (ii) interfacing with both the software product and the

corresponding policy server database, (iii) making a permission-to-run availability determination, with respect to local usage of the software product, on the basis of applicable data from the license server and the

5 corresponding policy server database, so that enforcement of license terms applicable to the software product at a given local node is achieved on the basis of both license policy maintained at such local node as well as applicable data from the license server.

10 In a further embodiment, each policy server database contains data specifying conditions under which usage of each of a plurality of software products is permitted on the node on which the database is maintained. Additionally, each policy server daemon interfaces with each software
15 product. In this manner, enforcement of license terms applicable to each software product at a given node is achieved on the basis of both locally maintained license policy and applicable data from the license server.

In a further embodiment, each node has a log file
20 maintained, in association with each policy server daemon, to record recent software product usage on that node. The policy server daemon is accordingly configured to handle instances when data from the license server is
25 unavailable -- for example, when the computer acting as the license server is non-operational or when the network is non-operational. In particular, the policy server daemon may permit a node to run a software product, in the absence of license server data, if the node's log file indicates a sufficient level of recent usage of the software product on
30 the node. The circumstances under which such a permission-to-run availability determination is favorable may be established by the node's policy server database.

In yet further embodiments, the policy server database and the log file may be encrypted. Furthermore the
35 interface between the policy server daemon and each software product may be made secure. When one or more of the software products are subject to concurrent licensing

restrictions specified in the policy server databases, the policy server daemon may be permitted to reserve a predetermined time interval over which the applicable node has a guaranteed opportunity to utilize a given software
5 product. The reservation is accomplished by having the node's policy server daemon communicate to the license server over the predetermined time interval that the node is using the given software product, regardless whether the software product is actually being used.

10 It can be seen that the present invention permits a single database at each node to specify all of the conditions under which the node may access any of the software products on the network. Furthermore, as described in further detail below, in order to invoke the licensing
15 administration function carried out in accordance with the present invention, each software product need contain only a simple and short segment including the instruction:

ILic-get_license

20 followed by parameters identifying license details for the particular software product. A branching routine (which may be made available to all the software products, and called by the particular software product after this instruction) then
25 specifies program flow depending on whether a license is available (the remainder of the program can be run) or not (the program operation is terminated and a message is displayed to the user).

Brief Description of the Drawings

30 The foregoing features of the invention will be more readily understood by reference to the following description taken with the accompanying drawings in which:

Fig. 1 is a block diagram showing operation of a preferred embodiment of the invention in a network;

35 Fig. 2 is a block diagram illustrating the interrelation of important modules of the embodiment;

Fig. 3 is a block diagram of the main logical flow of

the computer program used in the embodiment;

Fig. 4 is a block diagram of the main processing of a validated "get license" message;

Fig. 5 is a block diagram of the manner in which the
5 policy server database is structured;

Fig. 6 is a block diagram providing more detail than Fig. 4 of the logical flow of the processing of a "get license" message;

Fig. 7 is a block diagram of license acquisition
10 processing referred to as item 623 in Fig. 6;

Fig. 8 is a block diagram of clock message processing for licenses on the main list of licenses that have been established; and

Fig. 9 is a block diagram of clock message processing
15 for licenses moved to the recovery list by item 88 of Fig. 8.

Detailed Description of Specific Embodiments

The invention is applicable to computer networks of the type having an arrangement, such as NetLS, for tracking
20 software product usage, associated with one of the computers on the network acting as a license server. The present embodiment is described with respect to a Unix network; however, the software used by the license server in implementing such an arrangement, and the particular network
25 type, are a matter of design choice.

Fig. 1 shows the manner in which a preferred embodiment of the invention may be implemented on a Unix network. Each computer node 11 of the network may be running a variety of software products, such as PDS (item 12a), EMS (item 12b),
30 and so forth (shown through item 12j). Each of these products includes a call "get_license" to the local policy server daemon 16 for a determination whether a license is available to run the product in question. As used in this detailed description, the term "license" refers not to a
35 written document between the licensor and the licensee, but rather to the availability of permission to run the software product. The local policy server daemon 16 operates at the

- 9 -

computer node 11 and makes the permission-to-run availability determination by reference to its associated policy server database 14, also located at the node, to identify the rules specifying the circumstances under which a license would be granted. The daemon 16 also communicates over the network with the applicable license server. In this figure, three separate license servers are shown: one (item 18a) running NetLS; another (item 18b), SunLic; and another (item 18c), Highlands. The license server communicates with the daemon 16 using the applicable license software NetLS (item 17a), SunLic (item 17b), or Highlands (item 17c), and informs the daemon whether usage of the software product on the network is such that a license may be granted in accordance with the policy established by the database 14. If so, the daemon 16 reports the license to the applicable software product 12, and to the applicable license server 18. If there is no successful communication with the applicable license server 18, if the database 14 so permits, the daemon 16 will consult a log file 15 recording instances of recent software product usage, and if there has been a sufficient level of recent software product usage that has been licensed, the daemon will grant a temporary user license (TUL) to run the software product.

The communication between the applicable software product 12 and the local policy server daemon 16 is handled as an interprocess communication in Unix. Here the Unix "message" is used as the means of communication, but this is a matter of choice, and other means of communication, such as pipes or shared memories, may be used. In order to reduce to risk of tampering by the licensee with the license availability determination made by the policy server daemon 16, the rules database 14 and the log file 15 may be encrypted using techniques known in the art. Similarly, the message communication from the application to the policy server daemon 16 can be subject to validation using techniques known in the art to assure that the message is indeed from the pertinent software product.

The embodiment described herein has been implemented for use with a variety of types of licenses. (Numerous license types may be created and enforced by the invention, but the following types are illustrative.) One type is the concurrent use license. A concurrent use license is issued, from the server running the Licensing System (sometimes called the "license server" in this description and the claims following), with respect to a software product being used on a node in a network. The license server controls the levels of concurrent usage of the software product. Concurrent use licenses are returned to the license server when they are no longer needed. For example, if the Licensing System on the server permits five concurrent licenses for a given software product, then five users on the network can run the software product concurrently.

The concurrent use license is actually implemented as part of a two-tier structure. The first tier is a "base license," and the second tier is a "version-specific" license. The base license controls the number of simultaneous users of a software product. The version license controls the version of the software product that may be utilized by the user. The base license typically expires at the end of each year, and may be renewed. The version license typically never expires. The version license provides a mechanism for controlling how many base licenses are for a software product that is under a maintenance agreement. As an example, a user may have purchased a license to five copies of version A of a software product, but kept maintenance on only three copies. In such a case the user would receive five base licenses (which expire each year and were replenished unless the applicable computers were sold), plus five version A licenses that never expire. This user would subsequently receive only three version B licenses for the three copies under maintenance. Under such an arrangement, the user could still run five copies of version A of the software product, or a mix of version A and version B software as

- 11 -

long as the mix does not exceed five copies in total and does not exceed three copies of version B.

The policy server database file 14 of a computer node stores the license requirements for each software product to
5 be run at that node. For each software product, the database may identify the number of base license "tokens" and version license "tokens", obtained from the server running the Licensing System, that are necessary for operation of that software product on the particular
10 computer constituting the node. (The particular computer, for example, may be particularly fast in processing, and therefore a higher license fee may be required for running the software product on such computer, resulting here in a larger number of tokens required for the base and version
15 licenses.)

Another type of license is a node-locked license, which is tied to a particular computer node and cannot be used by other nodes. The node-locked license token is designated for a particular node when created. In a further variation
20 of the node-locked license, a "reserved" license may be established, that is, the policy server daemon may be permitted to reserve a predetermined time interval over which the applicable node has a guaranteed opportunity to utilize a given software product. (The reservation is
25 accomplished by having the node's policy server daemon communicate to the license server over the predetermined time interval that the node is using the given software product, regardless whether the software product is actually being used.)

30 A single use license can be used only for one invocation of the software product. Single use licenses are useful for emergency situations, peaks in usage, or demonstrations. A day use license is similar to a single use license, except that a day use license remains available
35 on the computer node that acquired it for 24 hours after the time of acquisition.

A temporary user license (TUL), described above, is

- 12 -

issued on a temporary basis when the server running the Licensing System becomes unavailable. A TUL is designed for emergency situations and is granted on a per user, per node, per software product, per usage history basis.

5 Fig. 2 illustrates the structure of a program implementing the embodiment of Fig. 1 for a single license server running one or more Licensing Systems, such as NetLS. The program is written in standard C. A communication
10 module 21 handles communication with the various software products, one of which is here shown as item 26. If the software product includes the "get_license" instruction, the communication module 21 refers to the licensing dispatch
15 module 22. The licensing dispatch module 22, by reference to the policy server database 14 and the applicable Licensing System, makes the license availability
20 determination. The Licensing System shown here is No. 1, and client portion 25 is accessed by licensing dispatch 22, which may access other Licensing Systems depending on the software product 26 and information in the policy server
25 database file 14. The client portion of the Licensing System 25 communicates over the network with the server portion 251. In the event that there is no successful communication with the server portion 26, the communication
30 module may trigger the temporary user license (TUL) module 27 to consult with the history log file 15 to determine if there is a sufficient level of recent licensed usage of the software product at this node to permit the grant of a
35 temporary user license (TUL). In any event, the communication module 21 reports the license availability determination by directing a message to the software product's process. The communication module is also responsible for sending a periodic signal (a "ping") to the license server to indicate continued use of a license. Another module 28 causes recordation of license usage in
40 license usage file 281 for reporting purposes. A file 252 of node-locked licenses is maintained locally. The communication module 21 is controlled by timer interval

- 13 -

handler 29, which in turn receives periodic signals from PS driver 291 that has been incorporated into the operating system.

Fig. 3 illustrates the main logical flow of the program carried out by the communication module 21. After initialization 31, the program gets the next message from any processes, and, in particular, from any software products that may be invoked from the node on which the program is running. Next the message is validated (item 10 33), and then the message is processed (item 34). After processing of the current message, the program loops to seek the next message again.

The most important message is "get_license", and this message is processed as shown in Fig. 4. The first step 15 41 is to determine the availability of a license. The license availability determination is made in the licensing dispatch module 22.

After the license availability determination is made as shown in step 41 of Fig. 4, if a license is granted, that 20 fact is reported to the software product in step 43. If the license is not granted because of a lost connection to the server running the Licensing System (determination in step 44), there is a check to see if usage of the software product is possible "under grace", that is, whether there 25 has been sufficient recent licensed usage of the software product at the node to permit granting of a TUL. If so, a TUL is granted (step 47). If not, or if the license was denied for reasons other than a lost connection, the program communicates (step 46) the fact of no license availability 30 to the software product.

Additionally, the communication module of the policy server daemon may reserve a predetermined time interval over which the applicable node has a guaranteed opportunity to utilize a given software product. The reservation is 35 accomplished by having the module communicate to the license server over the predetermined time interval that the node is using the given software product, regardless whether the

software product is actually being used.

The construction of the policy server database is shown in Fig. 5. License prices are initially established by management decision in price book 51, which forms the basis for assigning token values (step 52) required for license grant. The license cost to use a software product can also vary as a function of the hardware platform (i.e., the model of the computer) on which the product is running. Accordingly, the platform indicator data 54 and the rules defining the different types of licenses 53 all form a part of the structure of the policy server database 55. In order to assure integrity of the database, it is encrypted.

Fig. 6 is a block diagram providing more detail than Fig. 4 of the logical flow of the processing of a "get license" message. Initially (step 61), memory is allocated for the structure of the applicable license to be added to the list of license structures in memory. Unless the structure shows a reserved license (tested in block 62), the policy server database file 14 of Figs. 1 and 2 is accessed (step 621) to determine the applicable license terms. If access is successful (tested in block 622), then license acquisition processing (described in connection with Fig. 7) follows (step 623).

If, as a result of license acquisition processing, a license is granted (tested in block 625), the history log file 15 of Figs. 1 and 2 is then updated (step 631) to reflect this event. Thereafter, the policy server driver 291 of Fig. 2 is informed (step 63), the license usage file 281 of Fig. 2 is updated for use in generating later reports (step 64), the return status the operation is checked (step 641), and a status message is built and sent (step 65) to the software product that had included the "get license" call. If the return status is a failure (tested in step 641), the license structure is removed from memory (step 642) before sending the the status message to the software product.

If, as a result of the license acquisition processing

- 15 -

of step 623, a license has not been granted, the error messages produced by the Licensing System are analyzed to a single reason (step 626), and the return status for the software product is determined. If the embodiment described
5 herein is not in the enforcement mode (determined in step 627), then the return status is simply a warning (generated in step 643). If the embodiment is in the enforcement mode, there is a check (step 647) to determine if the connection with the license server is lost. If there is no lost
10 connection, the policy server database file 14 is checked (step 646) for the appropriate license failure conditions, and then the return status is determined (step 644). If there is a lost connection, processing follows (step 645), to determine on the basis of the history log file 15 and
15 data in the policy server database file 14 whether a TUL is available. If a TUL is available, the return status is a warning (step 643), as in the case when the system is not in the enforcement mode. Once the return status has been determined, processing is the same as if a license has been
20 granted; that is, the driver is informed, the license usage file is updated, the return status is checked and if necessary the license structure is removed from memory, and the appropriate status message is built and sent to the software product (steps 63, 64, 641, 642, and 65).

25 If after the determination (step 647) that there is a lost connection, and a TUL is not available (step 645), processing loops back to license acquisition processing (step 623) to attempt again to get a license from the license server. If the policy server database file 14 cannot be
30 successfully accessed in step 621 to determine the relevant license rules (a matter checked in step 622), the processing goes to determine (in step 627) whether the system is in the enforcement mode and to generate an appropriate return status. If in step 62, the license structure shows a
35 reserved license, access to the policy server database file 14 is skipped altogether, and the driver is informed (step 63) directly.

Fig. 7 is a block diagram of the license acquisition processing referred to as item 623 in Fig. 6. In accordance with this processing, there is first sought a "base" license token and then a "version" license token, where "base" and "version" have the meanings described above following the description of Fig. 1. Initially, the policy server database file 14 is cycled through to determine the enabled base token type (step 71)--for example node-locked, or concurrent access, or use once. The Licensing System on the server is then called (step 711) to seek the designated enabled token. If the base token is granted (checked at step 712), the policy server database file 14 is then cycled through to determine the enabled version token type (step 72). If the version token is granted (checked at step 722), the return is "license granted" (step 73). In each case if processing through the policy server database is not complete (checked for, in the case of the base token at step 713 and in the case of the version token at step 723), the database is cycled through again, the Licensing System is called to seek an enabled token, and there is a test to see if the token is granted. If the end of the list has been reached (tested at step 713 for the base token and 723 for the version token) and the applicable token has not been obtained, a failure is returned (step 725). If the base token has been granted, but the version token denied, then the base token is first freed (step 724) before the failure is returned in step 725.

Fig. 8 is a block diagram of clock message processing for licenses on the main list of licenses that have been established. First, a license is picked as part of a cycle through the main list of licenses in memory (step 81). Next there is a check whether a process exists for this license (step 82). If there is no process, the license is returned to the Licensing System, and associated housekeeping is done (step 821), and the program then picks the next license (step 81) to begin processing again. If it is determined that there is a process, then it is determined whether the

- 17 -

license needs to be "pinged" to satisfy requirements of the Licensing System to keep the license (step 83). The implementation here generates a ping every 10 minutes. If no ping is currently necessary, the program again picks the next license (step 81) to begin processing again. If a ping is necessary, it is sent (step 84), and if successful (i.e., the Licensing System reports that the license is still valid (tested in step 85), the program again picks the next license (step 81) to begin processing again.

10 If the ping is unsuccessful, a failure counter is incremented (step 86), and there is a test (step 87) to determine if the failure counter is above an established threshold. If it is, then the failure counter is cleared (step 88) and the license in question is moved to the recovery list (step 89). If it is not, then the program again picks the next license (step 81) to begin processing again.

Fig. 9 is a block diagram of clock message processing for licenses moved to the recovery list in step 89 of Fig. 8. First, a license is picked as part of a cycle through the recovery list of licenses in memory (step 91). Next there is a check whether a process exists for this license (step 911). If there is no process, any remaining part of the license is returned to the Licensing System, and associated housekeeping is done (step 94), and the program then picks the next license (step 91) to begin processing again. A check (in step 912) is made to determine whether the exit flag had been set in step 935, and if so, the process of the software product (application) is signalled to exit (step 913), the exit counter is decremented (step 914), and a test (step 915) is made to determine if the exit counter has reached zero. If so, the application process is killed (step 916). In either event, the next license is picked from the recovery list (91), and processing for the next license resumes as before.

If the exit flag had not been set, then a replacement license is sought (step 921), and a test (922) is made to

determine whether a license has been granted. If a replacement license has been granted, then

If a replacement license has not been granted, the replacement failure counter is incremented (step 93) and
5 then tested (step 931) to determine if it is above a threshold (here typically 3). If it is not above the threshold, then the next license is picked from the recovery list (91), and processing for the next license resumes as before. If it is above the threshold, the
10 policy server database file 14 is consulted (step 932) to determine whether running of the software product is permitted (step 933). If not, then the exit counter and exit flag are set up; if running is permitted, the replacement failure counter is decremented (step 934). In
15 either case, the next license is picked from the recovery list (91), and processing for the next license resumes as before.

Many other implementations of the invention described herein are possible. For example, the particular types of
20 licenses described here are merely examples. The use of base and version licenses are thus a matter of design choice. The manner in which the failure to obtain a license is handled can also be tailored to suit the policies of the licensor of the software products in question.

What is claimed is:

1. An improved system for administration, on a computer network, of license terms for use of a software product on the network, the system being of the type wherein the
5 network has a plurality of digital computers, each computer at a node, in communication with each other over a data path, and the system has usage tracking means, associated with one of the computers acting as a license server, for
10 (i) causing the storage of the number of licenses available for running the software product on nodes of the network,
(ii) identifying the current set of nodes with respect to which a license has been granted to run the software product at a given time, and (iii) determining whether at any given
15 time any licenses remain to be granted for permitting an additional node to run the software product, so that the software product may include instructions to cause enforcement of the license terms;

wherein the improvement comprises:

(a) a policy server database containing data
20 specifying conditions under which usage of the software product is permitted on any given node; and
(b) policy server means, maintained and operating locally as an independent process, on each computer, with respect to which the license terms are to be enforced, in
25 association with the policy server database, for (i) communicating with the license server, (ii) interfacing with both the software product and the policy server database, and (iii) making a permission-to-run availability
30 product, on the basis of applicable data from the license server and the policy server database, so that enforcement of license terms applicable to the software product at a given local node is achieved on the basis of both license
35 policy maintained in the policy server database as well as applicable data from the license server.

2. A system according to claim 1, wherein each computer at a node with respect to which license terms are to be

enforced includes means for maintaining locally a policy server database, containing data specifying conditions under which usage of the software product is permitted on such node.

5 3. A system according to claim 1, further comprising:

(c) log means for recording and maintaining a log file of recent software product usage on each computer at a node with respect to which license terms are to be enforced, such log file being accessible to such policy server means, and
10 wherein such policy server means includes means for making a permission-to-run availability determination in the absence of data from the license server on the basis of data from the policy server database and the log file, so that a favorable determination is possible if the log file
15 indicates a sufficient level of recent usage of the pertinent software product on the computer on which such policy server means is operating.

4. A system according to claim 2, wherein

(i) each policy server database contains data
20 specifying conditions under which usage of each of plurality of software products is permitted on the computer on which the database is maintained, and

(ii) each policy server means includes means for interfacing with each of the software products,

25 so that enforcement of license terms applicable to each software product at a given local node may be achieved on the basis of both license policy maintained at such local node as well as applicable data from the license server.

5. A system according to claim 4, further comprising:

30 (c) log means, maintained locally in association with each policy server means, for recording and maintaining a log file of recent software product usage on the computer on which such log means is maintained, such log file being accessible to such policy server means, and wherein such
35 policy server means includes means for making a permission-to-run availability determination in the absence of data from the license server on the basis of data from the policy

- 21 -

server database and the log file, so that a favorable determination is possible if the log file indicates a sufficient level of recent usage of the pertinent software product on the computer on which such policy server means is
5 operating.

6. A system according to claim 1, wherein the policy server database is encrypted.

7. A system according to claim 5, wherein the policy server database and the log file are encrypted.

10 8. A system according to claim 4, wherein the policy server means include means for maintaining a secure interface with each of the software products.

9. A system according to claim 7, wherein the policy server means includes means for maintaining a secure
15 interface with each of the software products.

10. A system according to claim 4, wherein one of the policy server databases includes a limit on the number of nodes that may simultaneously use a given software product and wherein the corresponding policy server means associated
20 with such policy server database includes reservation means for informing the license server, over a predetermined time interval, that the node associated with such policy server means is using the given software product, regardless whether such software product is actually being used, so
25 that such node will always be available to use such software product, despite attempts to use the software product at other nodes which if successful would otherwise foreclose use at such node of such software product, with the effect that the reservation means reserves use of such software
30 product at such node over the predetermined time interval.

11. A system according to claim 5, wherein one of the policy server databases includes a limit on the number of nodes that may simultaneously use a given software product and wherein the corresponding policy server means associated
35 with such policy server database includes reservation means for informing the license server, over a predetermined time interval, that the node associated with such policy server

means is using the given software product regardless of whether such software product is actually being used, so that such node will always be available to use such software product despite attempts to use the software product at other nodes which if successful would otherwise foreclose use at such node of such software product, with the effect that the reservation means reserve use of such software product at such node over the predetermined time interval.

12. A system according to claim 9, wherein one of the policy server databases includes a limit on the number of nodes that may simultaneously use a given software product and wherein the corresponding policy server means associated with such policy server database includes reservation means for informing the license server, over a predetermined time interval, that the node associated with such policy server means is using the given software product, regardless whether such software product is actually being used, so that such node will always be available to use such software product, despite attempts to use the software product at other nodes which if successful would otherwise foreclose use at such node of such software product, with the effect that the reservation means reserves use of such software product at such node over the predetermined time interval.

13. A computer network comprising:

(a) a plurality of digital computers, each computer at a node, in communication with each other over a data path;

(b) usage tracking means, associated with one of the computers acting as a license server, for (i) causing the storage of the number of licenses available for running the software product on nodes of the data path, (ii) identifying the current set of nodes with respect to which a license has been granted to run the software product at a given time, and (iii) determining whether at any given time any licenses remain to be granted for permitting an additional node to run the software product;

(c) a policy server database, maintained locally on such computer with respect to which it is desired to enforce

license terms applicable to usage of the software products, containing data specifying conditions under which usage of any given one of the software products is permitted on the computer on which the database is maintained; and

5 (d) policy server means, maintained and operating locally, on each computer with respect to which it is desired to enforce license terms applicable to usage of the software products, and in association with the corresponding policy server database, for (i) communicating with the
10 license server, (ii) interfacing with both (aa) each of the software products and (bb) the corresponding policy server database, and (iii) making a permission-to-run availability determination, with respect to local usage of any given software product, on the basis of applicable data from the
15 license server and the corresponding policy server database, so that enforcement of license terms applicable to the given software product at a given node is achieved on the basis of the license policy maintained at such local node as well as applicable data from the license server.

20 14. A computer network according to claim 13, further comprising:

(e) log means, maintained locally in association with each policy server means, for recording and maintaining a log file of recent software product usage on the computer on
25 which such log means is maintained, such log file being accessible to such policy server means, and wherein such policy server includes means for making a permission-to-run availability determination in the absence of data from the license server on the basis of data from the policy server
30 database and the log file, so that a favorable determination is possible if the log file indicates a sufficient level of recent usage of the pertinent software product on the computer on which such policy server is operating.

15. A computer network according to claim 14, wherein the
35 policy server database and the log file are encrypted.

16. A computer network according to claim 12, wherein the policy server means includes means for maintaining a secure

interface with each of the software products.

17. A computer network according to claim 15, wherein the policy server means includes means for maintaining a secure interface with each of the software products.

5 18. A digital storage medium encoded with instructions for a given computer in a computer network of the type having:

(i) a plurality of digital computers, each computer at a node, in communication with each other over a data path;

(ii) usage tracking means, associated with one of the
10 computers acting as a license server, for (i) causing the storage of the number of licenses available for running the software product on nodes of the network, (ii) identifying the current set of nodes with respect to which a license has been granted to run the software product at a given time,
15 and (iii) determining whether at any given time any licenses remain to be granted for permitting an additional node to run the software product,

the instructions when loaded into the given computer establishing:

20 (a) data structure for a policy server database, maintained locally on the given computer, containing data specifying conditions under which usage of any given one of the software products is permitted on the given computer; and

25 (b) policy server means, maintained and operating locally, on the given computer, and in association with the policy server database, for (i) communicating with the license server, (ii) interfacing with both (aa) each of the software products and (bb) the policy server database, and
30 (iii) making a permission-to-run availability determination, with respect to local usage of any given software product, on the basis of applicable data from the license server and the policy server database, so that enforcement of license terms applicable to the given software product at the given
35 computer is achieved on the basis of the license policy maintained at the given computer as well as applicable data from the license server.

- 25 -

19. A system, for administration of license terms for use of a software product on a computer, comprising:

(a) a policy server database containing data specifying conditions under which usage of the software product is permitted on the computer;

(b) policy server means, operating on the computer, in association with the policy server database, for (i) interfacing with the software product and the policy server database and (ii) making a permission-to-run availability determination, with respect to usage of the software product, on the basis of data from the policy server database.

15

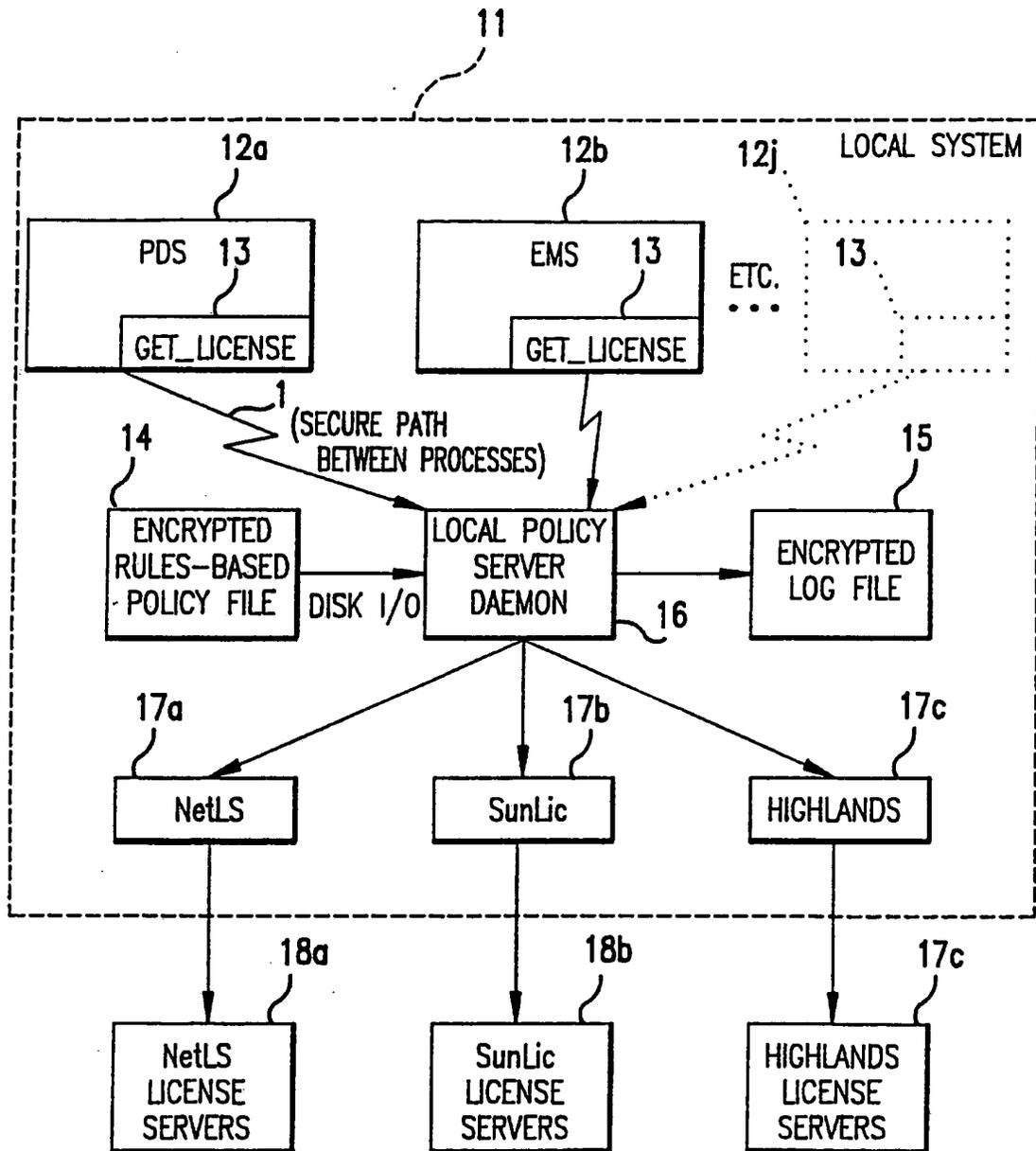
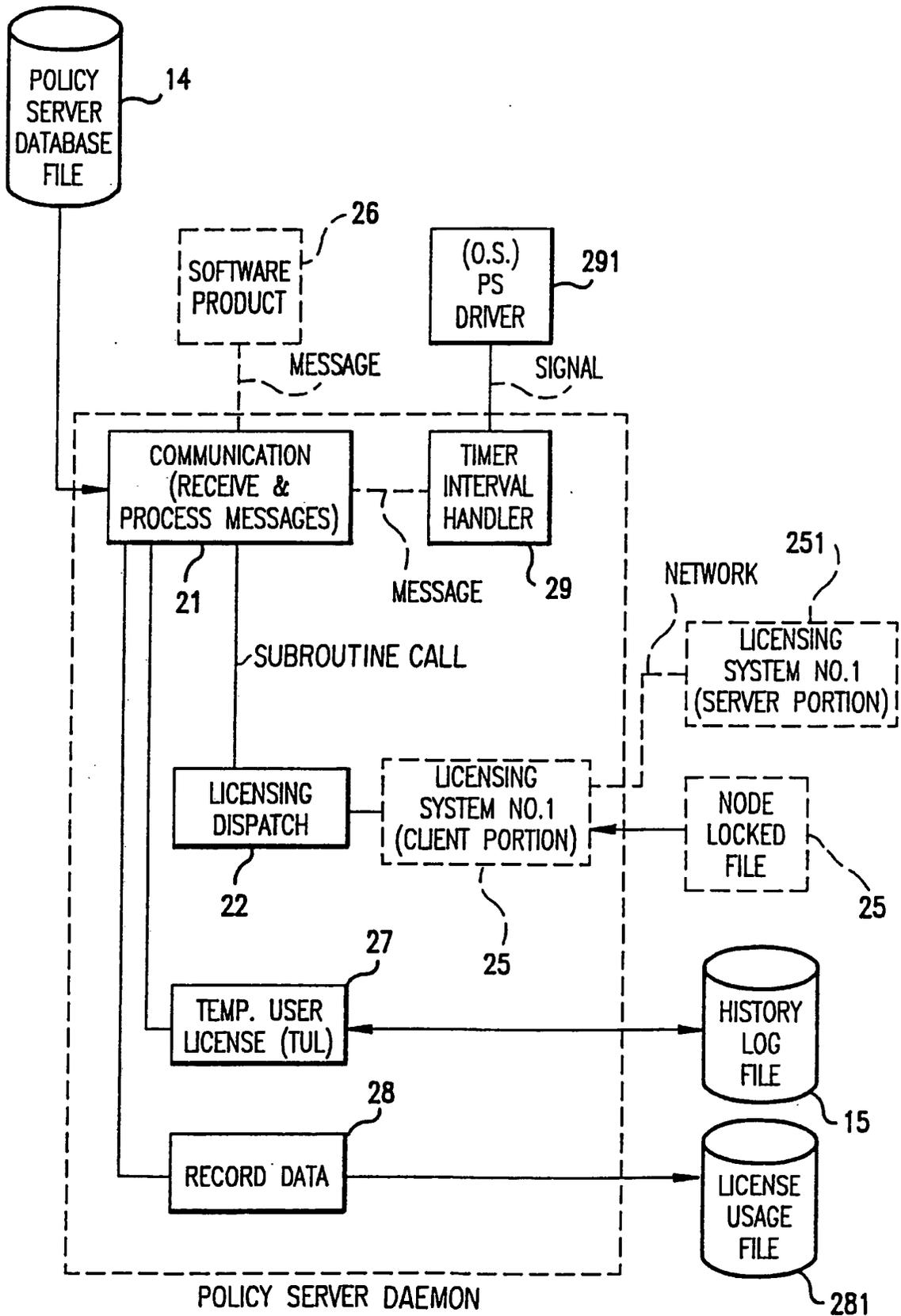


FIG. 1



SUBSTITUTE SHEET

FIG.2

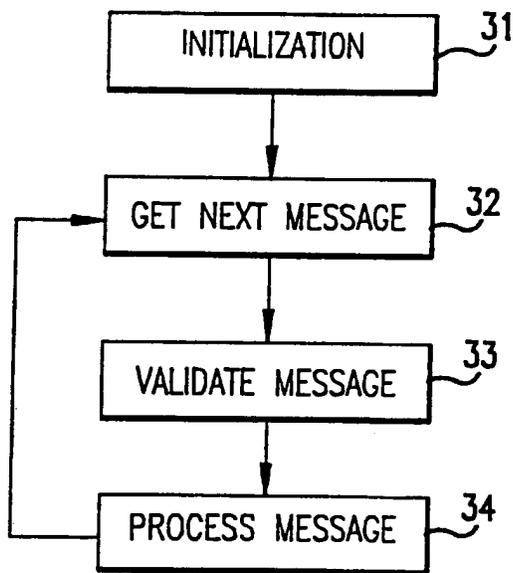


FIG.3

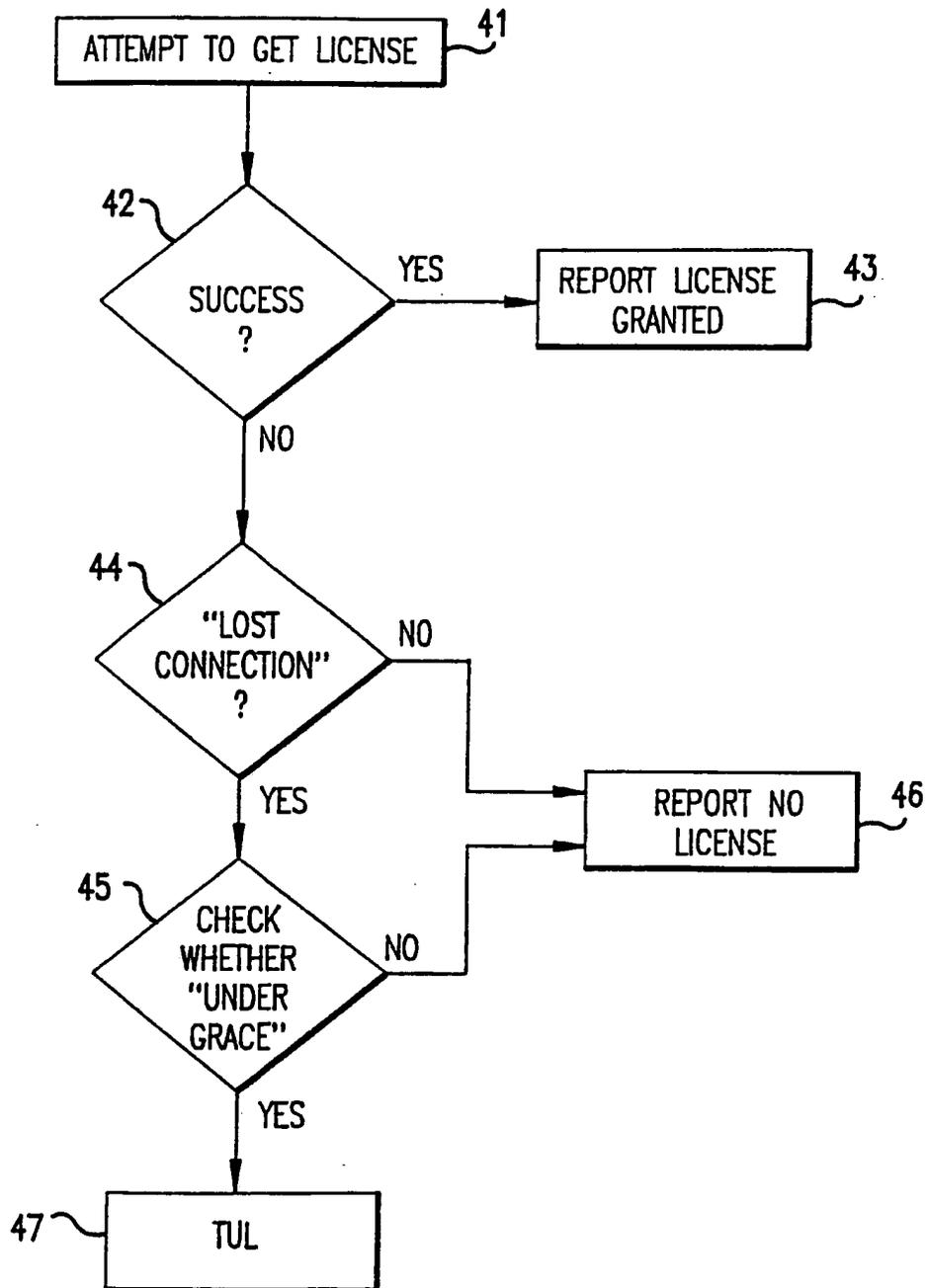


FIG.4

SUBSTITUTE SHEET

5/9

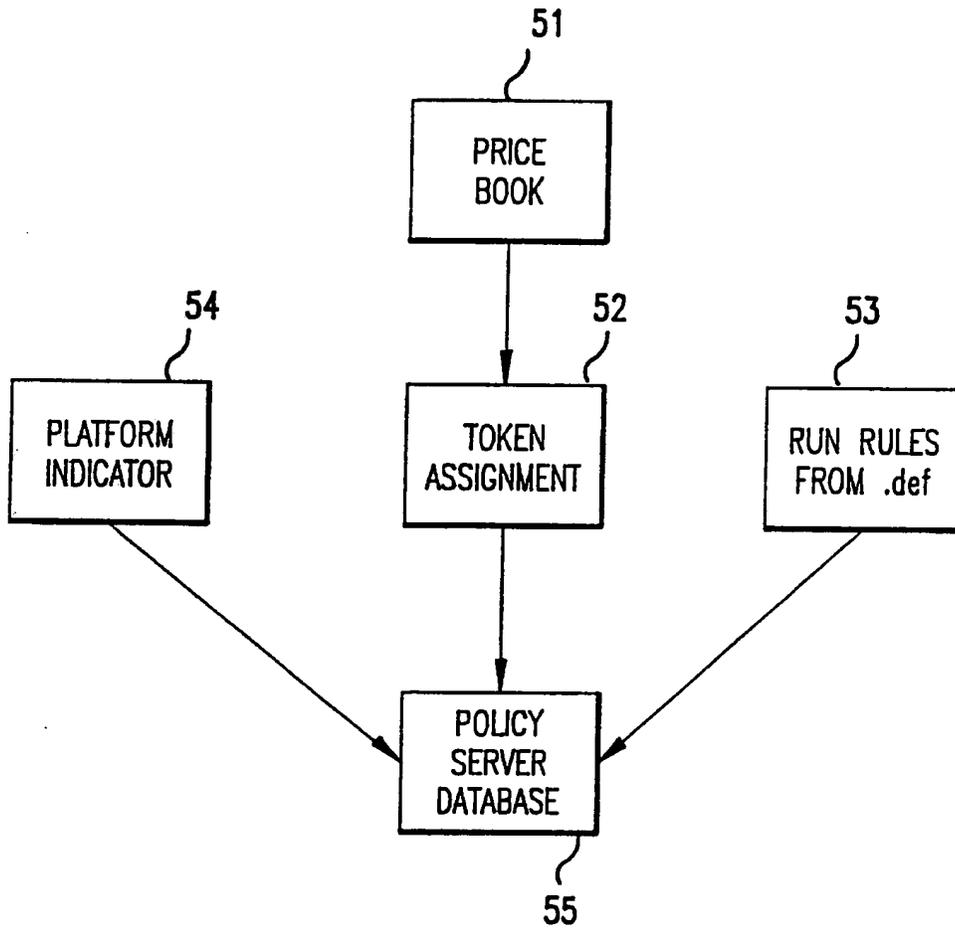


FIG.5

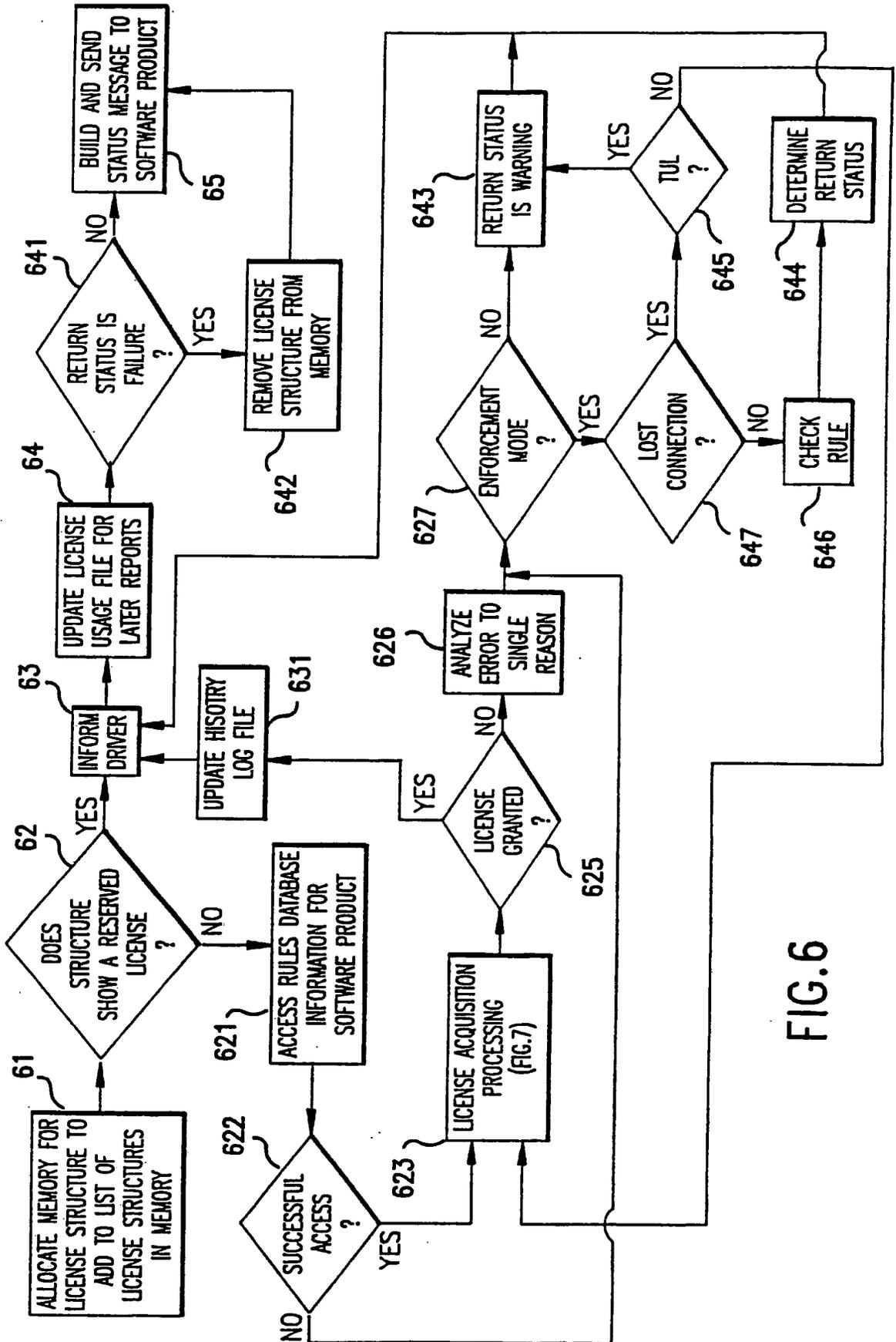


FIG. 6

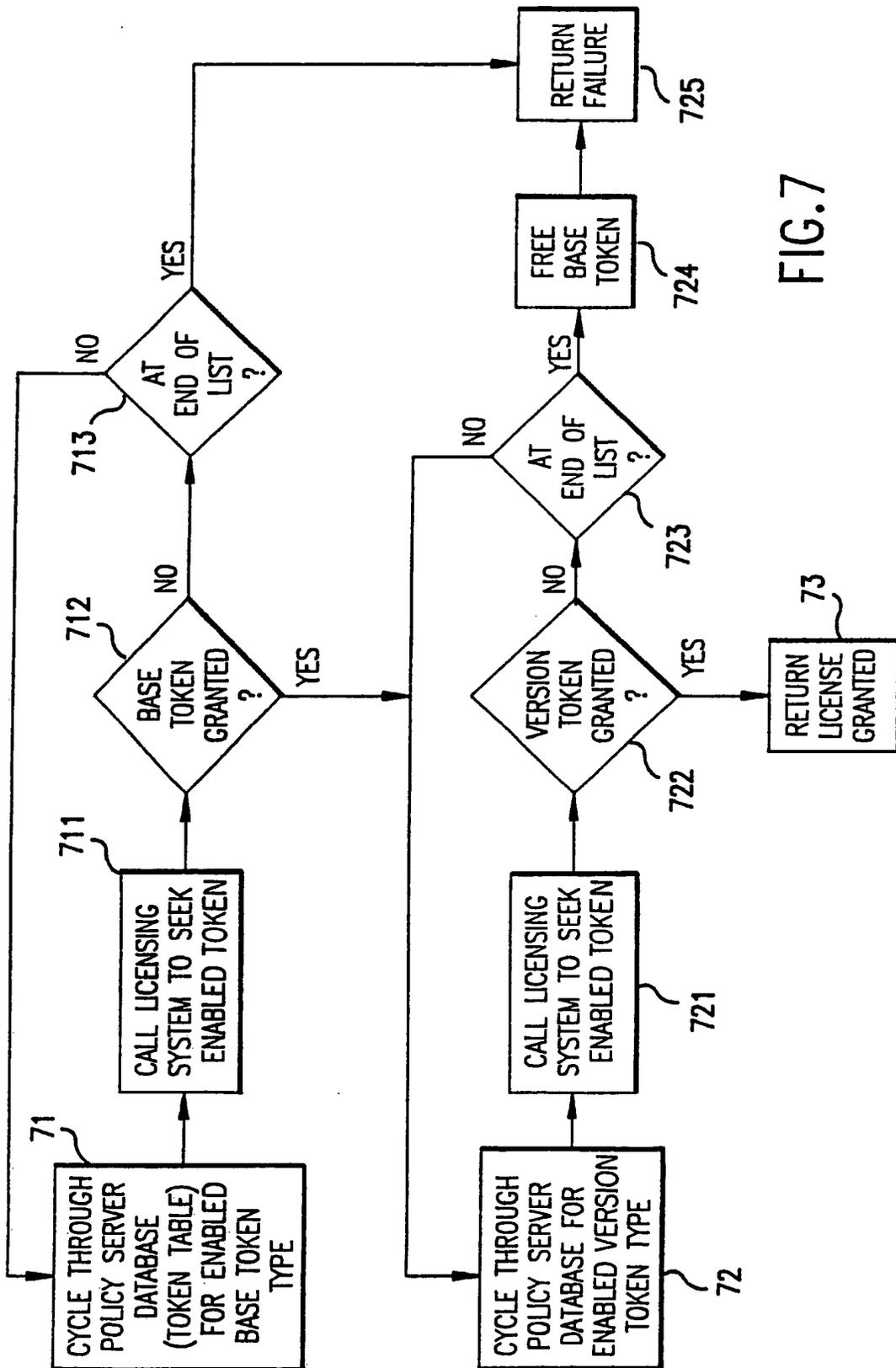


FIG. 7

SUBSTITUTE SHEET

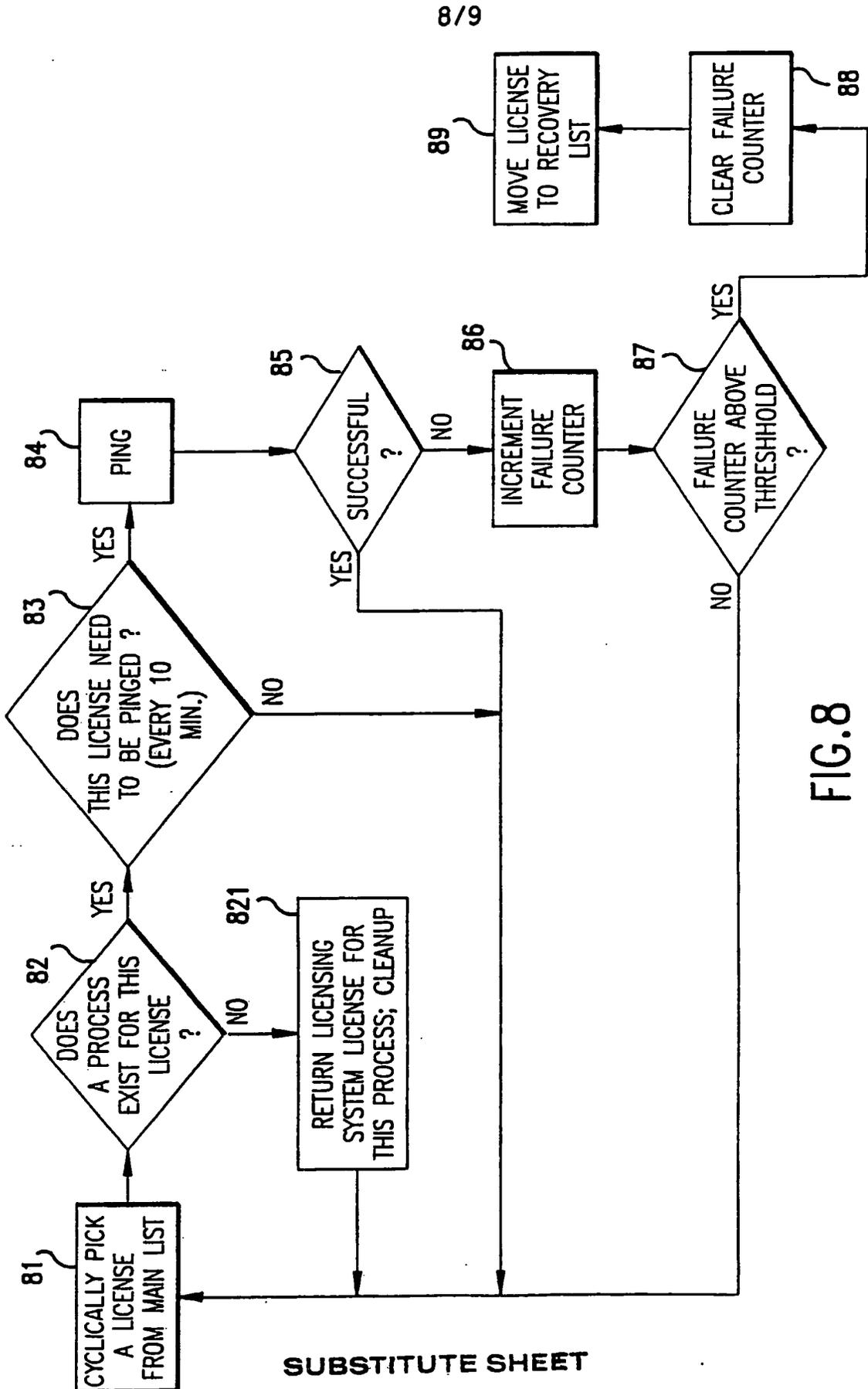


FIG.8

SUBSTITUTE SHEET

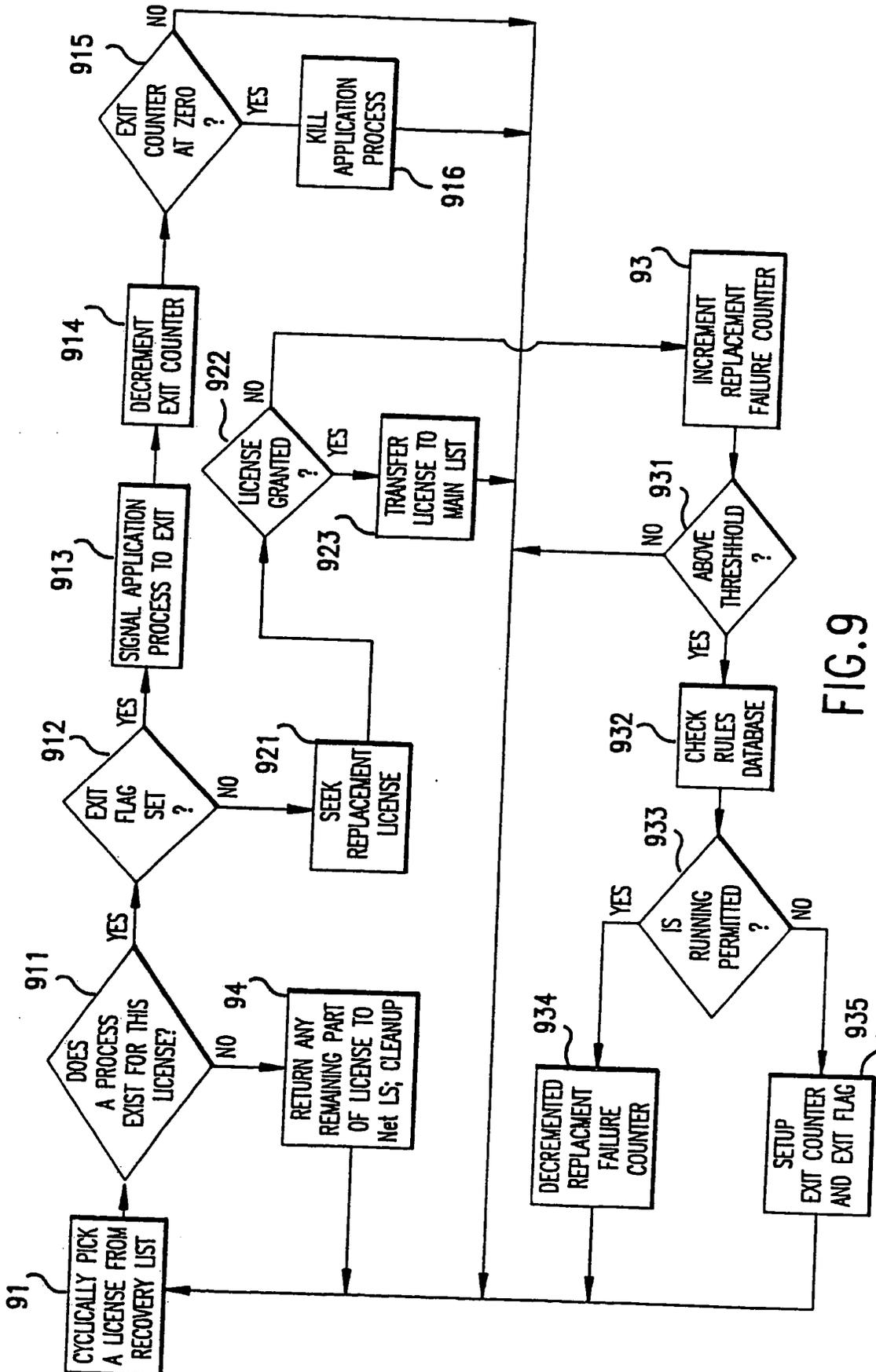


FIG.9

INTERNATIONAL SEARCH REPORT

PCT/US 92/10215

International Application No

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) ⁶		
According to International Patent Classification (IPC) or to both National Classification and IPC Int.Cl. 5 G06F1/00; G06F11/34		
II. FIELDS SEARCHED		
Minimum Documentation Searched ⁷		
Classification System	Classification Symbols	
Int.Cl. 5	G06F	
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched ⁸		
III. DOCUMENTS CONSIDERED TO BE RELEVANT⁹		
Category ¹⁰	Citation of Document, ¹¹ with indication, where appropriate, of the relevant passages ¹²	Relevant to Claim No. ¹³
A	GB,A,2 236 604 (SUN MICROSYSTEMS, INC.) 10 April 1991 see page 9, line 11 - page 10, line 28 see page 12, line 16 - page 13, line 13 see page 14, line 20 - page 16, line 27 see figures 1-3 ---	1-7, 13-15, 18-19
A	US,A,5 023 907 (APOLLO COMPUTER) 11 June 1991 see column 2, line 49 - column 5, line 42 see figures 1,2 --- -/--	1-5, 13-14, 18-19
<p>¹⁰ Special categories of cited documents : ^{"A"} document defining the general state of the art which is not considered to be of particular relevance ^{"E"} earlier document but published on or after the international filing date ^{"L"} document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) ^{"O"} document referring to an oral disclosure, use, exhibition or other means ^{"P"} document published prior to the international filing date but later than the priority date claimed</p> <p>^{"T"} later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention ^{"X"} document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step ^{"Y"} document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. ^{"&"} document member of the same patent family</p>		
IV. CERTIFICATION		
Date of the Actual Completion of the International Search 26 FEBRUARY 1993	Date of Mailing of this International Search Report 23.09.93	
International Searching Authority EUROPEAN PATENT OFFICE	Signature of Authorized Officer JOHANSSON U.C.	

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)		Relevant to Claim No.
Category °	Citation of Document, with indication, where appropriate, of the relevant passages	
A	EP,A,0 332 304 (DIGITAL EQUIPMENT CORP.) 13 September 1989 see column 3, line 31 - column 6, line 8 see column 6, line 42 - column 7, line 23 see column 8, line 33 - column 9, line 44 see figure 1 -----	1,2, 10-13, 18

Form PCT/ISA/210 (extra sheet) (January 1985)

**ANNEX TO THE INTERNATIONAL SEARCH REPORT
ON INTERNATIONAL PATENT APPLICATION NO.**

US 9210215
SA 67461

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information. 26/02/93

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
GB-A-2236604	10-04-91	US-A- 5138712	11-08-92
		CA-A- 2025434	03-04-91
		JP-A- 4100148	02-04-92
US-A-5023907	11-06-91	None	
EP-A-0332304	13-09-89	US-A- 4937863	26-06-90
		JP-A- 2014321	18-01-90

EPO FORM P007

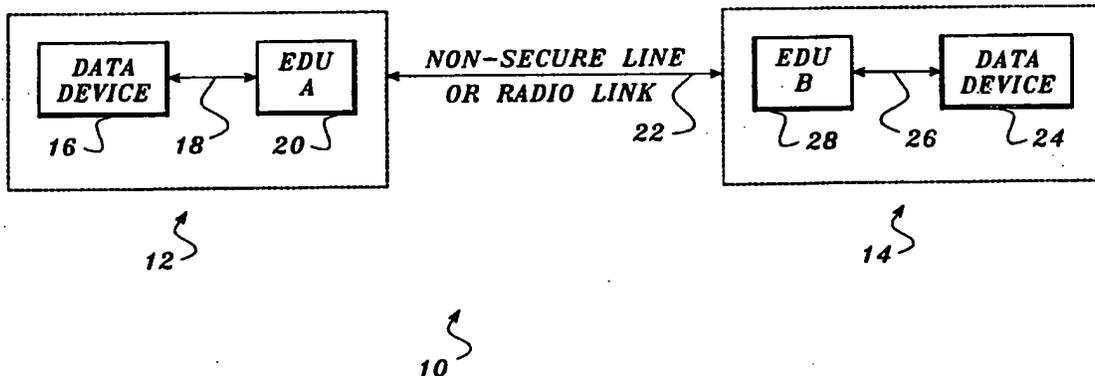
For more details about this annex : see Official Journal of the European Patent Office, No. 12/82



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(15) International Patent Classification 5 : H04L 9/00</p>	<p>A1</p>	<p>(11) International Publication Number: WO 94/03003 (43) International Publication Date: 3 February 1994 (03.02.94)</p>
<p>(21) International Application Number: PCT/US93/04340 (22) International Filing Date: 4 May 1993 (04.05.93) (30) Priority data: 07/917,598 23 July 1992 (23.07.92) US (71) Applicant: CREST INDUSTRIES, INC. [US/US]; 201 Frontage Road North, Suite B, Pacific, WA 98047 (US). (72) Inventors: RASMUSSEN, Harry, Ronald ; 6105-4th Street Court Northeast, Tacoma, WA 98422 (US). LaBOUNTY, Jack, Daley ; 16931 Southeast 32nd Place, Bellevue, WA 98008 (US). ROSENOW, Michael, James ; 1420 Northwest Gillman, Suite 2305, Issaquah, WA 98027 (US).</p>		<p>(74) Agent: ANDERSON, Ronald, M.; Christensen, O'Connor, Johnson & Kindness, 2800 Pacific First Centre, 1420 Fifth Avenue, Seattle, WA 98101-2347 (US). (81) Designated States: AT, AU, BB, BG, BR, CA, CH, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, KZ, LK, LU, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG). Published <i>With international search report.</i></p>

(54) Title: ENCRYPTION/DECRYPTION APPARATUS WITH NON-ACCESSIBLE TABLE OF KEYS



(57) Abstract

An encryption/decryption unit (EDU) that handles management of encryption keys used in the secure exchange of data over non-secure communication links. Each EDU includes a central processing unit (CPU) that controls its operation, random access memory (RAM) in which tables of key exchange keys (KEKs) are stored, and a data encryption standard (DES) coprocessor that implements a data encryption algorithm developed by the U.S. National Bureau of Standards - all comprising a module that is embedded in a potting material. Attempts to remove the potting material either by mechanical or solvent means are likely to result in loss of the data and program code stored in the module. The CPU includes special circuitry enabling it to operate in an encrypted mode so that it can not be interrogated to discover the program or data stored therein. This program enables the EDU (20) to establish secure communications with another similar EDU (28) over a non-secure link. Each EDU establishing a secure communications session randomly generates a portion of a session data encryption key (DEK) that is encoded by using a KEK from either a public or private table of keys stored in the embedded RAM. The two EDUs exchange the encrypted portions of the DEK, decrypt the portions, and then logically combine them to determine the current session DEK. Use of a stored EDU ID in each EDU comprising the link prevents a third EDU from bridging the link to tap into the communications between two stations.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NE	Niger
BE	Belgium	GN	Guinea	NL	Netherlands
BF	Burkina Faso	GR	Greece	NO	Norway
BG	Bulgaria	HU	Hungary	NZ	New Zealand
BJ	Benin	IE	Ireland	PL	Poland
BR	Brazil	IT	Italy	PT	Portugal
BY	Belarus	JP	Japan	RO	Romania
CA	Canada	KP	Democratic People's Republic of Korea	RU	Russian Federation
CF	Central African Republic	KR	Republic of Korea	SD	Sudan
CG	Congo	KZ	Kazakhstan	SE	Sweden
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovak Republic
CM	Cameroon	LU	Luxembourg	SN	Senegal
CN	China	LV	Latvia	TD	Chad
CS	Czechoslovakia	MC	Monaco	TC	Togo
CZ	Czech Republic	MG	Madagascar	UA	Ukraine
DE	Germany	ML	Mali	US	United States of America
DK	Denmark	MN	Mongolia	UZ	Uzbekistan
ES	Spain			VN	Viet Nam
FI	Finland				

**ENCRYPTION/DECRYPTION APPARATUS WITH NON-ACCESSIBLE
TABLE OF KEYS**

Field of the Invention

The present invention generally pertains to apparatus for encrypting and
5 decrypting data, and more specifically, to apparatus for implementing the encryption
and decryption process with secret encryption keys.

Background of the Invention

Procedures for encrypting and decrypting data for transmission over non-
secure radio or telephone links have been highly refined to meet the needs of the
10 military and industry. An encryption algorithm that is virtually unbreakable in any
reasonable time frame, by even the most powerful of high-speed computers, has been
developed and published by U.S. National Bureau of Standards and sanctioned for use
by industry in this country as an acceptable method for protecting computerized data
conveyed over non-secure channels. In fact, integrated circuits designed specifically
15 for encryption and decryption of data in accordance with this Data Encryption
Algorithm (DEA) are readily available from several vendors, such as Western
Digital™. The algorithm, like most encryption schemes, uses an encryption key to
encrypt data. Successful use of the DEA, and almost any other encryption/decryption
algorithm commonly employed, requires that the station receiving the encrypted
20 transmission have the same key used to encrypt the data in order to decrypt it.
Accordingly, no unauthorized party should know or have access to the encryption key
that is being used.

Unfortunately, for any prior art encryption/decryption system using the DEA
or similar algorithms, extensive security measures are required for managing and

periodically changing the encryption keys that are used. Any third party that gains access to the encryption key being used to encrypt data can tap into a non-secure line over which encrypted messages are transmitted and then use the key to decrypt messages that are intercepted. Even if knowledge of the encryption key used is limited to those operating the encryption/decryption equipment, there can be no assurance that others outside an organization will not breach security and learn the encryption key due to failure of someone in the organization to follow security procedures. As the size of a network over which secure communications must be maintained expands, the difficulty in managing the encryption keys used on the network grows exponentially.

Since any person with access to the encryption keys can breach the security of encrypted communications between members of the network, encryption keys must be changed on a regular basis. Frequent changes in the encryption keys in use minimizes the risk of disclosure by individuals that previously had access to the keys. However, any such change requires that the new encryption keys be distributed to all stations in the network. Typically, the new encryption keys are hand carried to each station site by bonded couriers; nevertheless, it is possible that a courier may compromise security. Even if a security breach does not occur, the cost of regularly distributing encryption keys to each station of a large network in this manner may be prohibitive.

For these reasons, it is preferable to use encryption keys at each station in a network that are not known to anyone, even those operating the encryption/decryption apparatus. Various techniques have been developed to access encryption keys stored in an electronic memory for this purpose. For example, a new encryption key can be selected for subsequent encryption of communications between stations based on the last encryption key that was used, by applying a secret formula to generate the new key. However, if the formula is discovered or otherwise becomes known by someone who is outside the organizational network, security of the encryption system is breached, since that person can generate the encryption keys that will subsequently be used, simply by applying the formula to any previously discovered key.

Clearly, it would be preferable to randomly generate the encryption key that is used to encrypt data transmitted to another station each time that communications are initiated. Yet, random generation of an encryption key at one station inherently renders the receiving station unable to decrypt the message, because it does not have the encryption key used. What is therefore required are means for transmitting the encryption key from one station to another in an encrypted form, with some provision

that enables the receiving station to decrypt the encryption key. Prior art encryption/decryption apparatus do not provide means to accomplish this task in an efficient manner that is not easily circumvented. Any key exchange key (KEK) that is used in the process of transferring an encryption key for encrypting and decrypting the message to the other station must be available to both stations, but can not be available to anyone outside the secure network of stations. Even if the encryption apparatus is available to someone outside the organization, it should be virtually impossible to discover the KEKs used by stations comprising the network, if secure communications are to be maintained.

10 The foregoing aspects and many of the attendant advantages of this invention over the prior art will become more readily appreciated as the same becomes better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings.

Summary of the Invention

15 In accordance with the present invention, encryption/decryption apparatus for ensuring secure communications between two stations include encryption processor means for encrypting and decrypting data using a session data encryption key (DEK) that is input thereto. Control means coupled to the encryption processor means are provided for controlling the operation of the encryption processor means. The control means supply the encryption processor means with the data for encryption and decryption and with an encryption key for use in encrypting and decrypting the data to produce an output signal in response to programmed instructions. These programmed instructions cause the control means to automatically randomly select a part of a session DEK and to combine it with another part of the session DEK received from the other station to determine the session DEK that will be used by the encryption processor means to encrypt data. Non-volatile memory means that are coupled to the control means store a plurality of key encryption keys that are used by the encryption processor means in encrypting a part of the session DEK for transmission to the other station. The control means select the key encryption key from the plurality of key encryption keys as a function of a check value determined with the part of the session key.

30 Within the non-volatile memory means is disposed an internal power source that provides electrical power to maintain storage of the plurality of key encryption keys. Potting means encapsulate the encryption processor means, the control means, and the non-volatile memory means in a radio and light wave opaque material that is sufficiently hard and resistant to dissolution by solvents to prevent its removal without

damage to interconnections coupling the non-volatile memory means to the control means and damage to interconnections supplying electrical power to the non-volatile memory means from the internal power source. Such damage causes erasure of the plurality of key encryption keys stored in the non-volatile memory means. In addition, the control means respond to any attempt to externally interrogate the non-volatile memory means by causing erasure of the key encryption keys stored therein.

Multiplexer means are coupled to the control means to receive a data signal and a select signal therefrom, and are also coupled to the encryption processor means, an output port, and the memory means; the multiplexer means selectively convey the data signal to one of the encryption processor means, the output port, and the memory means, in response to the select signal. The control means include a non-volatile memory for retaining program steps and a unique identification code that identifies a specific encryption/decryption apparatus. In addition, the control means include means for locking the control means and its non-volatile memory to prevent data and program steps from being read externally after storage of the program steps in the non-volatile memory is complete. The means for locking include means for encrypting data and memory addresses defining memory storage locations within the non-volatile memory of the control means and within the non-volatile memory means.

Brief Description of the Drawings

FIGURE 1 is a block diagram of a communications network comprising two stations, each provided with an encryption/decryption unit (EDU) in accordance with the present invention, thereby enabling the stations to establish secure communications over a non-secure line or radio link;

FIGURE 2 is a schematic block diagram of one of the EDUs shown in FIGURE 1;

FIGURE 3 is a flow chart illustrating the logical steps implemented at one station by the EDU in selecting and encrypting a first portion of a session encryption key for transmittal to another station;

FIGURE 4 is a flow chart illustrating the logical steps implemented by the EDU at the other station in decrypting the first portion of the session encryption key, and in selecting and encrypting a second portion of the session encryption key for transmittal to the one station; and

FIGURE 5 is a flow chart illustrating the logical steps implemented by the EDU at the one station to decrypt the second portion of the session encryption key.

Detailed Description of the Preferred Embodiment

As noted above, one of the more difficult problems in establishing and maintaining an encrypted communication network is distributing secure DEKs to each station in the network on a regular basis. In FIGURE 1, a simple network for carrying out encrypted communications is shown generally at reference numeral 10. Network 10 is shown simply as two stations, including a station 12 and a station 14, but it will be appreciated that the network can comprise many other such stations.

Both stations 12 and 14 use similar components for encrypting and decrypting communications. For example, station 12 includes a data device 16, which may, for example, comprise a facsimile machine or personal computer (neither shown separately). Data device 16 is connected through lines 18 to an EDU A 20. Station 12 uses EDU A 20 to establish secure communications over a non-secure line (or radio link) 22 with station 14, which includes an EDU B 28. EDU B 28 is connected to a data device 24 over lines 26. Data device 24 is the same type of device as data device 16. Thus, if data devices 16 and 24 are facsimile machines, communications network 10 permits secure communication of facsimile information in an encrypted form between stations 12 and 14 over non-secure line 22.

Because of the manner in which secure communications are established between EDU A 20 and EDU B 28, tapping into non-secure line 22 using a similar EDU (not shown) would NOT enable a third party to breach secure communications between stations 12 and 14. In the preferred form of the present invention, communications between EDU A 20 and EDU B 28 are carried out using a session encryption key that is changed with each session and comprises two parts, one part randomly selected by EDU A 20, and the other part randomly selected by EDU B 28. Thus, the present invention comprises the EDU at each of the communicating stations 12 and 14. In establishing secure communications between two stations 12 and 14, the EDU at each station randomly select its respective portion of the session encryption key, encrypts that portion of the session encryption key, and transmits the encrypted respective portion of the session encryption key to the other station. Once both station 12 and station 14 have decrypted the portion of the session encryption key developed by the other station, the two portions are logically combined at each station to produce the complete or final session encryption key used for encrypting data transmitted between stations 12 and 14 during the current session. In addition, the EDUs are preprogrammed to ensure that the intended station in a two-way communication link is actually receiving or transmitting the encrypted data, to guard against a third party tapping into non-secure line 22 with another EDU. The EDUs

also ensure that the two portions of the session encryption key that are exchanged between stations 12 and 14 are correctly received and decrypted, thereby protecting against data errors that might have arisen in the transmission of the encrypted portions of the session encryption key between the two stations or in their decryption.

5 A block diagram of EDU 20 is shown in FIGURE 2; EDU 28 is exactly the same, except for having a different EDU identification number stored within it. EDU 20 includes a potted module 30 and an external input/output (I/O) bus 32 for providing interconnections between the EDU and the data device (or to other
10 apparatus) that will provide the data to be encrypted or will receive the data that is decrypted by the EDU. Module 30, which comprises virtually the entire EDU, is encapsulated within a radio opaque and light opaque potting compound 34 to prevent discovery of the internal circuitry and to prevent forced electromagnetic or visual
15 tapping, monitoring, or other forms of penetration that might be attempted to uncover encryption keys and other information included therein. The potting compound is sufficiently hard and resistant to abrasion to prevent its removal without damaging the components comprising the EDU or at least causing loss of important data stored therein. Of greatest sensitivity to maintaining the security of communications between
20 EDUs comprising a network is the need to protect against discovery of KEKs that are encrypted using a key that is unique to each EDU and is assigned to it when it is initialized. The encrypted KEKs are stored as tables within each EDU and are utilized for encrypting portions of the session encryption key that are exchanged between two stations and subsequently logically combined at each EDU to produce a session DEK that is used for encryption of data exchanged over non-secure line 22. To avoid
25 breaching the security of communications on network 10, it is absolutely imperative that these KEKs not become publicly known.

 In the preferred form of module 30, two sets or tables of KEKs are stored in encrypted form in a random access memory (RAM) 42. One set is called a "public" set, since each EDU that will be sold will include this set. The other set is a "private"
30 set of KEKs, which optionally may be randomly generated by a user for distribution to and storage in those EDUs comprising a private network of stations. The significance of the KEKs will be apparent from the description that follows. Any attempt to expose the internal circuitry of module 30 by use of a chemical, solvent, or mechanical means in order to access RAM 42 electronically or physically so as to access these
35 data will cause loss of the KEKs that are stored therein. RAM 42 preferably comprises a Dallas Semiconductor™ type DS 1213 smart socket in which is installed

a memory integrated circuit (not separately shown) comprising 128K x 8 bits of storage, i.e., yielding 1,048,576 bits of non-volatile static RAM organized as 131,072 words by 8 bits. This memory integrated circuit is a dual in-line package (DIP) package configuration of generally conventional design, but the smart socket contains an internal battery supply (not separately shown) sufficient to maintain data integrity in the absence of externally applied power for a period in excess of 10 years. Dallas Semiconductor also supplies an integrated circuit non-volatile memory device that includes an integral internal battery supply, and this type of device can be used in place of the smart socket and more conventional memory device combinations. In the event a chemical solution is used to dissolve potting compound 34 in an attempt to discover the KEKs stored in RAM 142, the material comprising RAM 142 (smart socket or memory device that includes the integral internal battery supply) will also be dissolved, thereby disconnecting the internal battery supply and erasing the KEKs stored therein.

Operation of module 30 to establish and conduct secure communications is controlled by a CPU 36, which includes 32K of embedded RAM (not separately shown). In the preferred embodiment, a Dallas Semiconductor™ type DS 5000 microchip integrated circuit is used for CPU 36. The DS 5000 integrated circuit includes non-volatile embedded RAM (not separately shown) and all information and programming stored therein are preserved in the absence of an externally applied voltage for up to 10 years. In addition, the internal data registers and key configuration registers of the DS 5000 integrated circuit are non-volatile. Data stored within the embedded RAM that comprise program steps carried out by CPU 36 in establishing secure communications can be modified after encapsulation of module 30 has been accomplished with potting material 34; however, initial loading of the embedded RAM within the DS 5000 microchip comprising CPU 36 is accomplished with a conventional universal asynchronous receiver/transmitter (UART) interface (not shown) that is connected through external I/O bus 32 by lines 76. In addition, control lines 50 connect CPU 36 to external I/O bus 32 and convey write, read, interrupt, and signals for ports 0-3 (P1.0-P1.3) of the CPU.

Data lines (D0-D7) 54 interconnect CPU 36 with RAM 42 and with a buffer 46. Buffer 46 comprises an SN 74HCT245 octal bus transceiver with a three-state output that is used to block external access to internal data transfers occurring within module 30, thereby preventing an external device from accessing KEKs stored in RAM 42 and other data transferred between components of the module. Buffer 46 is enabled via control signals supplied over a line 74 by CPU 36 when it is appropriate

to allow bi-directional data transfer to and from external I/O bus 32 through lines 52, and therefore to and from an external device.

To provide additional security, CPU 36 operates in an encrypted mode. The encrypted mode is deactivated prior to the initial loading of program steps and data into the embedded RAM of CPU 36. Before the initial loading of program code and data begins during manufacture of the EDU, a 40-bit encryption mode key is selected for use by CPU 36 in the encrypted mode. A data encryptor circuit and an address encryptor circuit (neither separately shown) within CPU 36 respectively control the form in which the program code is stored in the embedded RAM of the CPU and the addresses at which it is stored. As the initial loading of program steps is performed, the data encryptor circuit uses the 40-bit encryption mode key to transform or encrypt opcodes, operands, and data bytes at each memory location defined by the software. Similarly, the address encryptor circuit uses the encryption mode key in a different encryption algorithm to translate or encrypt a logical address of each data byte location into an encrypted address at which the data are actually stored. The contents of the embedded RAM are then verified, and the encrypted mode is enabled by setting a security lock bit. After the security lock bit is set to enable operation in the encrypted mode, the contents of the CPU's embedded RAM is unintelligible to an observer that might attempt to tap into its circuitry to discover the program code and other data stored therein. The address and data encryptor circuitry provides real time translation or decryption of program code and address locations to CPU 36 during subsequent operation of the EDU. Only program code and data stored in the CPU's embedded RAM that does NOT affect secure operation of the EDU can be changed after the security lock bit is set. Any attempt to externally interrogate the CPU to discover the 40-bit encryption key causes its erasure, rendering the contents of the embedded RAM useless. Even if the encrypted program code and data are thereafter read back from the embedded RAM in CPU 36, they can not be decrypted without the 40-bit encryption mode key, which is lost.

CPU 36 selects a specific storage location for a KEK within RAM 42 by setting 16 address bits. Lines 58 connect CPU 36 to a latch 44, and lines 60 connect latch 44 to RAM 42. To minimize the total number of pins required on CPU 36, the first eight address bits (A0-A7) and eight bits of data (D0-D7) use the same pins on CPU 36. These address bits and data are alternatively passed between CPU 36, latch 44, and RAM 42 over lines 58 and 60, respectively. The eight most significant bits of the address are conveyed on lines 56b directly from CPU 36 to RAM 42 and to external I/O bus 32. The least significant eight address bits (A0-A7) are carried on

lines 56a. In the preferred embodiment, the 16 address bits are available on lines 56 at external I/O bus 32 to address the embedded RAM in CPU 36 when it is initially loaded or subsequently modified.

Although CPU 36 controls the operation of module 30, the actual encryption and decryption of data is implemented by a data encryption standard (DES) coprocessor 38. DES coprocessor 38 is designed to encrypt and decrypt 64-bit blocks of data using the algorithm specified in the Federal Information Processing Data Encryption Standard (No. 46). A transfer rate of 807 kilobytes per second is implemented by DES coprocessor 38 under the control of a 10 megahertz clock circuit 48, to which the DES coprocessor is connected through lines 70. Data are transferred between CPU 36 and DES coprocessor 38 over lines 72. In the preferred embodiment, a Western Digital™ type DES WD20C03A integrated circuit is used for DES coprocessor 38, although other such devices are available from other suppliers. A decoder/multiplexer (MUX) 40 is connected through lines 68 to DES coprocessor 38 and through lines 66 to CPU 36. Decoder/MUX 40 is a three-line to eight-line circuit that decodes one of eight lines, dependent upon three binary select inputs and three enable inputs. Lines 66 carry the three binary select signals and the output signal from decoder/MUX 40 and line 68 carries selectable input 7. In addition, lines 62 carry selectable inputs 5 and 6 from RAM 42, while lines 64, which extend between decoder/MUX 40 and external I/O bus 32 convey selectable inputs 0-4.

The embedded non-volatile RAM in CPU 36 is loaded with the appropriate program steps for controlling the operation of EDU 20 at the time module 30 is manufactured. In addition, RAM 42 is loaded with a set of 65,535 public KEKs that are randomly generated from over 72 quadrillion possibilities. Each EDU that is thus produced stores the same table of 65,535 randomly generated public encryption keys. Any EDU can establish secure encrypted communications with any other EDU using the public KEKs. Also stored in RAM 42 is a user-generated table of over 65,535 randomly generated private encryption keys. These private KEKs are used for initiating secure communications with another EDU in the private network that has the same table of private KEKs stored within its RAM 42.

The steps involved in establishing secure communications between two EDUs are shown in FIGURES 3, 4, and 5. Not shown are any handshaking steps necessary to connect two EDUs in communication with each other so that data for a specific device can be transmitted between them. Preferably, such handshaking steps are

implemented by transmitting predefined data blocks between the two devices, but do not necessarily require action by either EDU.

In FIGURE 3, a flow chart 100 identifies the steps taken by EDU B 28, acting as the intended recipient, to establish secure communications. It will be apparent the steps in flow chart 100 could also be carried out by EDU A 20; however, the choice was made in the preferred embodiment to have the receiving station start the process of determining a session data encryption key, thereby avoiding the possibility that a third party posing as another station might tap into the unsecured line with an EDU to initiate the secure communications link. The method begins with a start block 102. In a block 104, EDU B 28 generates a 64-bit random data encryption key 1 (DEK1), which is one of over 72 quadrillion possible data encryption keys (i.e., all possible combinations of 56 bits).

The DEK1 is the first portion of a session data encryption key that will be subsequently used for transmitting encrypted data between the two EDUs. In a block 106, EDU B 28 then uses the DEK1 as the encryption key in implementing the DEA to encrypt one block of data. The use of the DEA to encrypt a single block of data is referred to as an electronic code book (ECB) method and is carried out by DES coprocessor 38 under the control of CPU 36. The ECB method employs the key (DEK1) to encrypt a 64-bit zero function, i.e., a function comprising 64 logical zeros, the result being used to determine a check value.

In a block 108, a KEK table entry value KEK1 comprising the 16 least significant bits (LSBs) of the 64-bit check value from block 106 is determined. The EDU uses the public or private table for KEKs, as specified by EDU A 20 during the handshaking that preceded establishing the secure communications link. The public table and private table of KEKs each represent a linear array of data, that can be taken in groups of four 16-bit words or 64-bits at a time, to define a KEK. The 16 LSBs of the check value determine the starting point or table entry value in the selected table to determine the 64 bits used as a KEK, as indicated in a block 110. Using the 64-bit KEK selected from the table as the encryption key, the EDU encrypts the value DEK1 using the ECB method in a block 112. A cyclic redundancy check (CRC) value for the KEK table that was selected is then determined in the conventional manner.

In a block 114, the EDU encrypts the KEK table CRC, its own EDU ID number (which is stored in within module 30 and is not user modifiable), and the KEK1 entry value using a predefined header encryption key and the ECB method to produce an encrypted key header. The header encryption key is stored in the embedded RAM within CPU 36 at the time that its programming is initially loaded

and is the same for each EDU. In a block 116, the EDU transmits the encrypted key header and encrypted DEK1 to EDU A 20, which initiated the communication. Although both parts of this transmission are encrypted, they are encrypted at different levels of security, since the encrypted key header is always sent encrypted with the same predefined (although inaccessible) key and the encrypted DEK1 uses a different key with virtually every communication session between two EDUs. The method for establishing secure communications continues with the other EDU, at a block B1 118.

In FIGURE 4, a flow chart 120 shows the steps carried out by EDU A 20 (the EDU that initiated the communication). Flow chart 120 begins at block B1 118 and proceeds to a block 122 wherein the encrypted key header and encrypted DEK1 received from EDU B 28 are parsed. In a block 124, the encrypted key header is decrypted using the predefined header encryption key with the ECB method, enabling the EDU to determine the KEK table CRC, the encoded EDU ID number of the EDU that transmitted the encrypted header, and KEK1.

A decision block 126 causes the CPU to determine if the KEK table CRC is correct, thereby ensuring that the KEK table used to encrypted the header is the same as the KEK table that will be used by EDU A 20. This step prevents two EDUs from attempting to communicate if they are using different private KEK tables or if the public table in used by one has become corrupted or is different than the normal public table of KEKs for some other reason. If the CRC value does not match the expected value, a block 128 stops communication between the EDUs. Under most circumstances, however, the KEK table CRC is correct and the logic proceeds to a block 130.

In block 130, EDU A 20 determines the 64-bit KEK that was previously selected from the public or private table by EDU B 28, using the KEK1 value that it just received as an offset to enter the table. The 64-bit KEK is then used with the ECB method to decrypt the value DEK1, as shown in a block 132.

In a block 134, a validity check is made to ensure that the decryption process was carried out correctly and that the encrypted data were not affected by noise or other problems during transmission. The validity check is carried out by using the decrypted DEK1 value and the ECB method to encrypt the 64-bit zero function. The result provides a check value, the 16 LSBs of which are a value KEK1'. The accuracy of the encryption/decryption process and transmission is confirmed in a decision block 136 if the EDU determines that KEK 1 equals KEK 1'. If not, a block 138 provides for indicating that an error has occurred in establishing secure communications, which leads to a stop block 140.

On the other hand, assuming that KEK 1 equals KEK 1', a block 142 directs the EDU to generate a 64-bit random value, DEK2, which is the second portion of the session data encryption key that will be used to encrypt data transmissions between the two EDUs. In a block 144, EDU A 20 performs a logical XOR to combine the first portion of the session key, DEK1, and the second portion, DEK2, to determine the final session data encryption key DEK.

In a block 146, DEK2 is used with the ECB method to encrypt the 64-bit zero function in order to determine a second check value. Using the 16 LSBs of the check value in a block 148, the EDU determines a table entry value KEK2. By entering the specified public or private table at the address offset determined by KEK2, four consecutive 16-bit words comprising a 64-bit KEK are determined in a block 150. The EDU uses the value of KEK from the table and the ECB method to encrypt DEK2 in a block 152.

With the predefined header encryption key, the EDU A 20 encrypts the KEK table CRC, its own EDU ID, and the table entry value KEK2, producing an encrypted key header in a block 154. The encrypted key header just produced and the encrypted DEK2 will be transmitted to EDU B 28 only if the next test is passed in a decision block 155.

Decision block 155 now determines whether the EDU ID that was decrypted from the header received from EDU B 28 in block 124 matches that of the EDU that was initially called, i.e., confirms that the intended recipient has responded. Since the encryption of the EDU ID is carried out automatically by EDU B 28, and can not be modified or affected by external signals, it is virtually impossible for a third party to use another EDU to break into a communications link and take part in establishing secure communications, since the encrypted EDU ID that is returned to the station that initiated the communication would then not match the expected EDU ID. A negative response to decision block 155 causes the process for establishing secure communications to be halted at a stop block 157. Otherwise, the process for establishing a secure communications link proceeds to a block 156. Block 156 provides for transmitting the encrypted key header and encrypted DEK2 to the other EDU, i.e., to EDU B 28, which is the intended recipient for subsequent encrypted communications. Thereafter, the logic proceeds to a block A2 158 in FIGURE 5.

FIGURE 5 illustrates a flow chart 160 defining the steps next implemented by EDU B 28. Following block 158, a block 162 provides for parsing the encrypted key header and encrypted DEK2. The encrypted key header is then decrypted in a block 164 using the ECB method in connection with the predefined header encryption

key, enabling EDU B 28 to determine the KEK table CRC, the EDU ID of the transmitting station, and the KEK2 table entry value. In a decision block 166, EDU B 28 determines if the KEK table CRC value is correct, i.e., confirms that the public or private table of KEKs used by EDU A 20 is the same as that being used by EDU B 28. If not, the communication process is halted at a block 168. Otherwise, the process continues with a block 170.

Block 170 provides for selecting a 64-bit KEK from the designated table of KEKs using the entry value KEK2 as an offset. In a block 172, the EDU uses the selected KEK value in connection with the ECB method to decrypt the encrypted DEK2. It then performs a validity check in a block 174, by using the DEK2 value in connection with the ECB method to encrypt the 64-bit zero function, thereby determining a check value and a table entry value KEK 2' that is based upon the 16 LSBs of the check value. A decision block 176 causes CPU 36 to determine if the decrypted KEK2 equals KEK2' that was just determined in block 174. If not, a block 178 provides for indicating that an error has occurred, leading to a stop block 180.

However, assuming that the validity check has a positive response, in a block 182, the EDU logically XORs DEK1 and DEK2 to determine the value of DEK for this session. At this point, both the receiving and transmitting station EDUs have established the current session data encryption key DEK. Before the communication session can proceed, one final check is made in a decision block 183.

Decision block 183 determines if the EDU ID sent by EDU A 20 in the key header that was decrypted in block 164 by EDU B 28 matches an expected EDU ID. If not, block 180 stops the process of establishing secure communications between the two EDUs. Decision block 183 thus determines if a third EDU has been used to intercept communications between EDU A 20 and EDU B 28; if not, the communication of encrypted data proceeds at a block 184.

The session DEK is used in a block 184 by EDU A 20 to encrypt data (such as facsimile or computer data) for transmission to EDU B 28, which then decrypts it using the same DEK. When EDU B 28 determines that the last of the data to be transmitted has been received and decrypted, a block 186 provides for resetting both EDUs to await the next communication. Thereafter, a stop block 188 terminates further communication between the two stations.

During the process of establishing secure communications, neither of the EDUs linking together transmits DEK1 or DEK2 in the clear. Either the public or private table of KEKs is used for encrypting the first and second portions of the

current session DEK. Consequently, only another EDU provided with the same control program and the same table of KEKs (producing the same CRC value) would be able to decrypt either the encrypted first or second portions of the DEK. Since the software program controlling the operation of the EDUs requires that the EDU ID number of the stations be encrypted as part of the key header information that is exchanged, a third EDU cannot be used to surreptitiously substitute for the intended receiving station or transmitting station during the establishment of the secure communication link. Consequently, only the two EDUs at the receiving and transmitting stations comprising a link are able to communicate to establish a session DEK and thereafter carry on secure communications.

Only an EDU having the same session DEK used to encrypt data can decrypt the data. Furthermore, although any EDU can establish secure communications with any other EDU using the public table of KEKs, only EDUs having the same private table of KEKs (determined from the KEK table CRC value) can establish a session DEK to communicate with each other. As a result, a corporation that generates its own table of private KEKs can ensure that secure communications are initiated only with other stations comprising its private network that include the same table of private KEKs.

While the DES algorithm is used in the preferred form of the present invention, it will be appreciated that other encryption algorithms that use an encryption key can also be employed. Further, when determining a check value, a predefined function other than the zero function can be used. It should also be apparent that the encrypted key header need not include the EDU ID, if a lower level of security is acceptable, for example, in a local network of EDUs exclusively using private KEKs. These and other modifications to the present invention will be apparent to those of ordinary skill in the art. Accordingly, it is not intended that the invention be in any way limited by the description of the preferred embodiment and modifications thereto, but instead that the scope of the invention be determined entirely by reference to the claims that follow.

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. Encryption/decryption apparatus for ensuring secure communications between two stations, said encryption/decryption apparatus disposed at each station comprising:

(a) encryption processor means for encrypting and decrypting data using an encryption key that is input thereto;

(b) control means, coupled to the encryption processor means, for controlling the operation of the encryption processor means, said control means supplying the encryption processor means with data for encryption and decryption and with the encryption key for use in encrypting and decrypting the data to produce an output signal in response to programmed instructions that cause it to automatically randomly select a part of a session data encryption key for use by the encryption processor means to encrypt data when combined with another part of the session data encryption key received from the other station; and

(c) non-volatile memory means, coupled to the control means, for storing a plurality of key encryption keys used by the encryption processor means in encrypting the part of the session data encryption key for transmission to the other station, said control means selecting the key encryption key from said plurality of key encryption keys as a function of a check value determined by the control means with the part of the session key.

2. The encryption/decryption apparatus of Claim 1, wherein said non-volatile memory means include an internal power source that supplies electrical power to maintain storage of the plurality of key encryption keys.

3. The encryption/decryption apparatus of Claim 1, further comprising potting means for encapsulating the encryption processor means, the control means, and the non-volatile memory means in a radio and light wave opaque material, said potting means being sufficiently hard and resistant to dissolution by solvents to prevent its removal without causing damage to interconnections coupling the non-volatile memory means to the control means and damage to interconnections supplying electrical power to the non-volatile memory means from the internal power source, such damage causing erasure of the plurality of key encryption keys stored in

the non-volatile memory means, said control means also responding to any attempt to externally interrogate the non-volatile memory means by causing erasure of the key encryption keys stored therein.

4. The encryption/decryption apparatus of Claim 1, further comprising multiplexer means, coupled to the control means to receive a select signal therefrom, and coupled to the encryption processor means, an output port, and the memory means, for selectively conveying a data signal thereto in response to the select signal.

5. The encryption/decryption apparatus of Claim 1, wherein the control means include a non-volatile memory for storing the programmed instructions and for storing a unique identification code that identifies a specific encryption/decryption apparatus.

6. The encryption/decryption apparatus of Claim 5, wherein the control means include means for locking the control means and its non-volatile memory to prevent data and program steps from being read externally or changed after storage of the programmed instructions in said non-volatile memory is complete.

7. The encryption/decryption apparatus of Claim 6, wherein the means for locking include means for encrypting data and memory addresses defining memory storage locations within the non-volatile memory of the control means and within the non-volatile memory means.

8. Encryption/decryption apparatus for ensuring secure communications, comprising:

(a) processor means for randomly selecting a partial session data encryption key;

(b) encryption means for encrypting the partial session data encryption key, producing an encrypted part key and decrypting another partial session data encryption key selected at another location; and

(c) means for conveying the encrypted part key to an output port so that it can be transmitted to the other location and for conveying an encrypted signal from an input port, said encryption means decrypting the other partial session data encryption key received from the other location as the encrypted signal, said

processor means combining the partial session data encryption key to determine the current session data encryption key that is subsequently used by it to encrypt data transmitted to the other location and to decrypt encrypted signals received from the other location.

9. The encryption/decryption apparatus of Claim 8, further comprising memory means for storing a plurality of key encryption keys, wherein the encryption means select a specific key encryption key from the plurality of key encryption keys as a function of a check value, said encryption means encrypting a predefined set of characters with said part of the encryption key to determine the check value.

10. The encryption/decryption apparatus of Claim 9, wherein the means for transmitting also transmit the check value determined by the encryption means.

11. The encryption/decryption apparatus of Claim 10, wherein the decryption means use a check value received from said other location to determine a specific key encryption key that was used to encrypt the other partial session data encryption key.

12. The encryption/decryption apparatus of Claim 11, wherein:

(a) the encryption means use the other partial session data encryption key decrypted by the decryption means to encrypt the predefined set of characters, producing a test value;

(b) said processor means compare the test value with a check value received from the other location and detect an error if the test value differs from said check value received from the other location; and

(c) if an error is detected in (b), said processor means halt communications with said other location.

13. The encryption/decryption apparatus of Claim 9, wherein said memory means store a unique identification code for that apparatus.

14. The encryption/decryption apparatus of Claim 9, wherein the memory means comprise a non-volatile memory circuit including an internal power source, said internal power source supplying electrical current to the non-volatile memory circuit to retain data stored therein, said memory means being encapsulated in a material that precludes physical inspection of the memory circuit, preventing discovery of the data stored therein, further comprising means for interrupting electrical current supplied from the internal power source to the memory circuit so that the data stored therein are erased if the material encapsulating the memory means is removed therefrom.

15. The encryption/decryption apparatus of Claim 8, wherein the processor means comprise a central processing unit that is programmed to control the encryption means and the decryption means according to a predefined set of instructions.

16. The encryption/decryption apparatus of Claim 8, wherein the encryption means comprise an integrated circuit that implements encryption and decryption of data from a plurality of sources in response to signals from the processor means, using the current session data encryption key, in accordance with a predefined encryption algorithm and a corresponding predefined decryption algorithm.

17. The encryption/decryption apparatus of Claim 9, wherein the memory means store a plurality of sets of key exchange keys, further comprising means for selecting one of the sets of key exchange keys from which the specific key exchange key is determined.

18. Encryption/decryption apparatus for ensuring secure communications, comprising:

(a) a sealed circuit encapsulated in a material opaque to radio and light waves, said sealed circuit comprising:

(i) a central processing unit that receives and transmits data in both an encrypted and decrypted form;

(ii) a memory circuit coupled to the central processing unit, at least one predefined set of key exchange keys being stored in the memory circuit,

said key exchange keys stored in the memory circuit being externally inaccessible, both physically by inspection and by downloading through the central processing unit;

(iii) an encryption/decryption coprocessor coupled to the central processing unit to receive data therefrom, said encryption/decryption coprocessor encrypting and decrypting the data under control of the central processing unit based upon a specified encryption key, the encryption/decryption coprocessor selectively generating a second set of key exchange keys that are also stored in the memory circuit;

(b) connector means for interconnecting the sealed circuit with external data input and output lines, the encryption/decryption coprocessor selectively encrypting the second set of key exchange keys and the connector means conveying the second set of key exchange keys in an encrypted form to an external device for distribution to other encryption/decryption apparatus comprising a limited network, whereby only encryption/decryption apparatus comprising the limited network can securely communicate with each other using the second set of key exchange keys, but can securely communicate with other like encryption/decryption apparatus that do not comprise the limited network using the predefined set of key exchange keys.

19. The encryption/decryption apparatus of Claim 18, further comprising memory means coupled to the central processing unit, for storing program steps controlling automatic determination of a session data encryption key for use in encrypting and decrypting data, said session data encryption key being determined in part by the central processing unit logically combining a first randomly selected portion of the session data encryption key that is received in an encrypted form from another location with a second randomly selected portion of the session data encryption key that the central processing unit transmits to the other location in an encrypted form.

20. The encryption/decryption apparatus of Claim 19, wherein one of the predefined set and the second set of key exchange keys is selectively used for encrypting said other portion of the session data encryption key.

21. The encryption/decryption apparatus of Claim 18, wherein the memory circuit stores a unique identification code for the sealed circuit that can not be changed, said central processing unit halting operation of the sealed circuit if data are received from the other location that specify a different identification code, thereby

preventing secure communications with an unintended encryption/decryption apparatus.

22. Encryption/decryption apparatus for ensuring secure communications between two stations, comprising:

(a) first processor means at one of the stations for randomly selecting a first part encryption key and second processor means at the other of the two stations for randomly selecting a second part encryption key;

(b) encryption means at said one station for encrypting the first part encryption key, producing an encrypted first part key;

(c) means for transmitting the encrypted first part key to said other station;

(d) decryption means at said other station for decrypting the encrypted first part key to determine the first part encryption key;

(e) encryption means at said other station for encrypting the second part encryption key, producing an encrypted second part key;

(f) means for transmitting the encrypted second part key to said one station; and

(g) decryption means at said one station for decrypting the encrypted second part key to determine the second part encryption key, said first processor means at said one station and said second processor means at said other station then combining the first part encryption key and the second part encryption key to determine an encryption key that is used to encrypt and decrypt subsequent communications between the two stations.

23. The encryption/decryption apparatus of Claim 21, further comprising memory means for storing a plurality of key encryption keys at each of the two stations, wherein the encryption means at each station select a specific key encryption key from the plurality of key encryption keys as a function of a first check value and a second check value, respectively, said encryption means at said one station encrypting a predefined set of characters with said first part encryption key to determine the first check value, and said encryption means at said other station encrypting the predefined set of characters with said second part encryption key to determine said second check value.

24. The encryption/decryption apparatus of Claim 22, wherein the means for transmitting from each station also transmit the respective first or second check value determined by the encryption means at each station.

25. The encryption/decryption apparatus of Claim 23, wherein the decryption means at said other station use the first check value received from said one station to determine the specific key encryption key that was used by the encryption means at said one station to encrypt the first part encryption key, and wherein the decryption means at said one station use the second check value received from said other station to determine the specific key encryption key that was used by the encryption means at said other station to encrypt the second part encryption key.

26. The encryption/decryption apparatus of Claim 24, wherein:

(a) the encryption means at said other station uses the first encryption key decrypted by the decryption means to encrypt the predefined set of characters, producing a test check value;

(b) said second processor means compares the test check value with the first check value and detects an error if the test check value differs from the first check value;

(c) the encryption means at said one station uses the second encryption key decrypted by the decryption means to encrypt the predefined set of characters, producing a test check value;

(d) said first processor means at said one station compares the test check value with the second check value and detects an error if the test check value differs from the second check value; and

(e) if an error is detected in (b), said second processor means halt communications with said one station, and if an error is detected in (d), said first processor means halt communications with said other station.

27. The encryption/decryption apparatus of Claim 22, wherein said memory means at each station store a unique identification code for that station.

28. The encryption/decryption apparatus of Claim 26, wherein the encryption means at said one station encrypt the unique identification code of said other station, the means for transmitting then transmitting an encrypted identification code to said other station, said decryption means at said other station decrypting the unique identification code.

29. The encryption/decryption apparatus of Claim 27, wherein said second processor means compare the decrypted unique identification code with the unique identification code stored in the memory means and if not identical, halt communications with said one station.

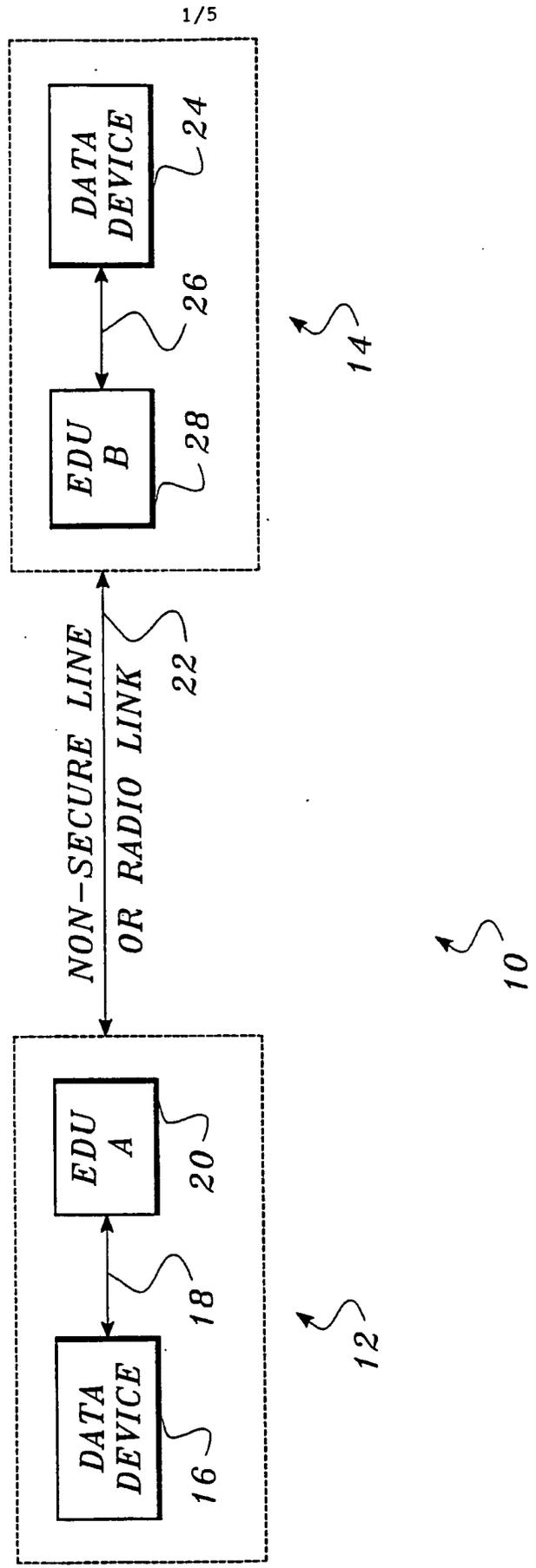


FIG. 1.

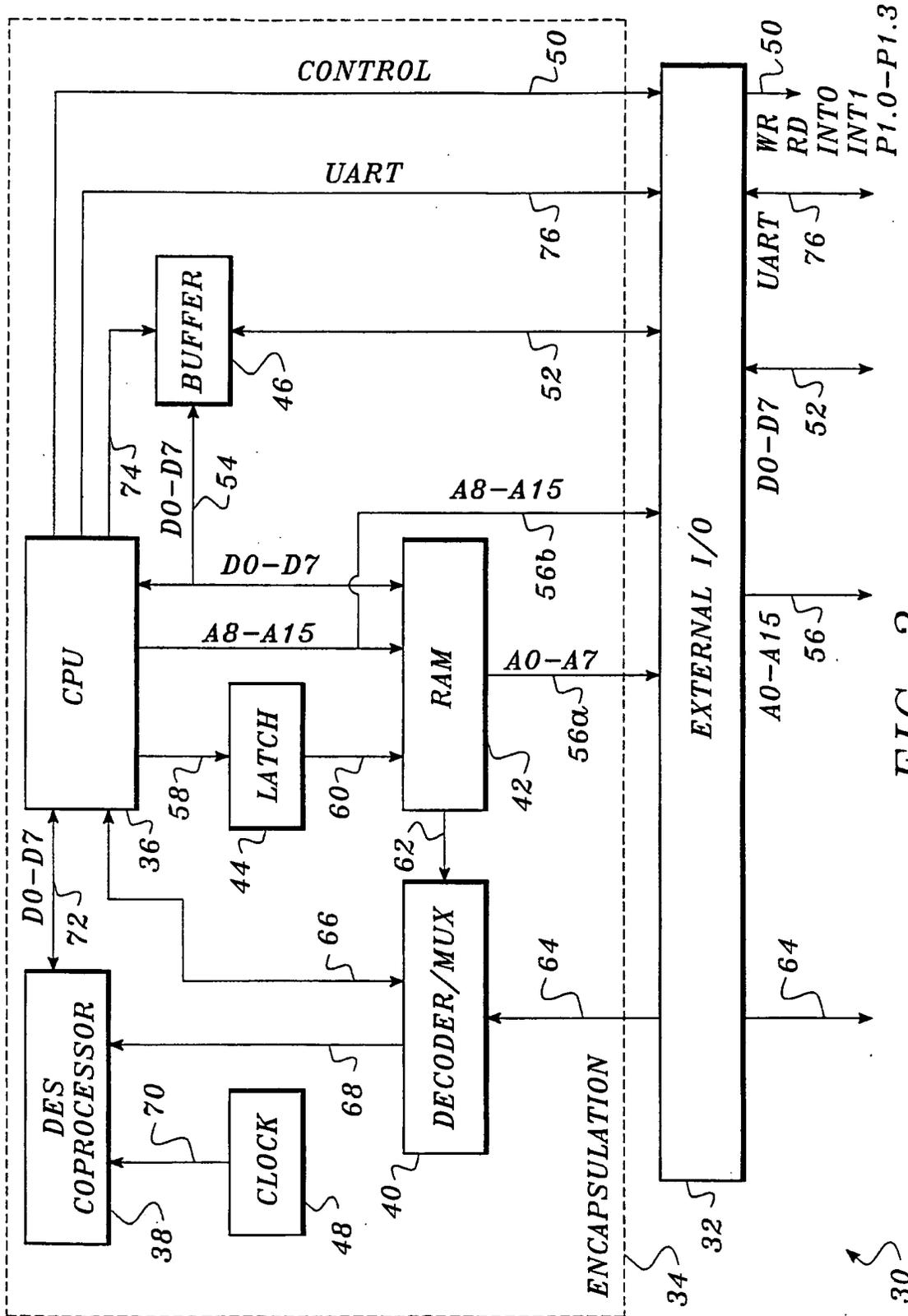
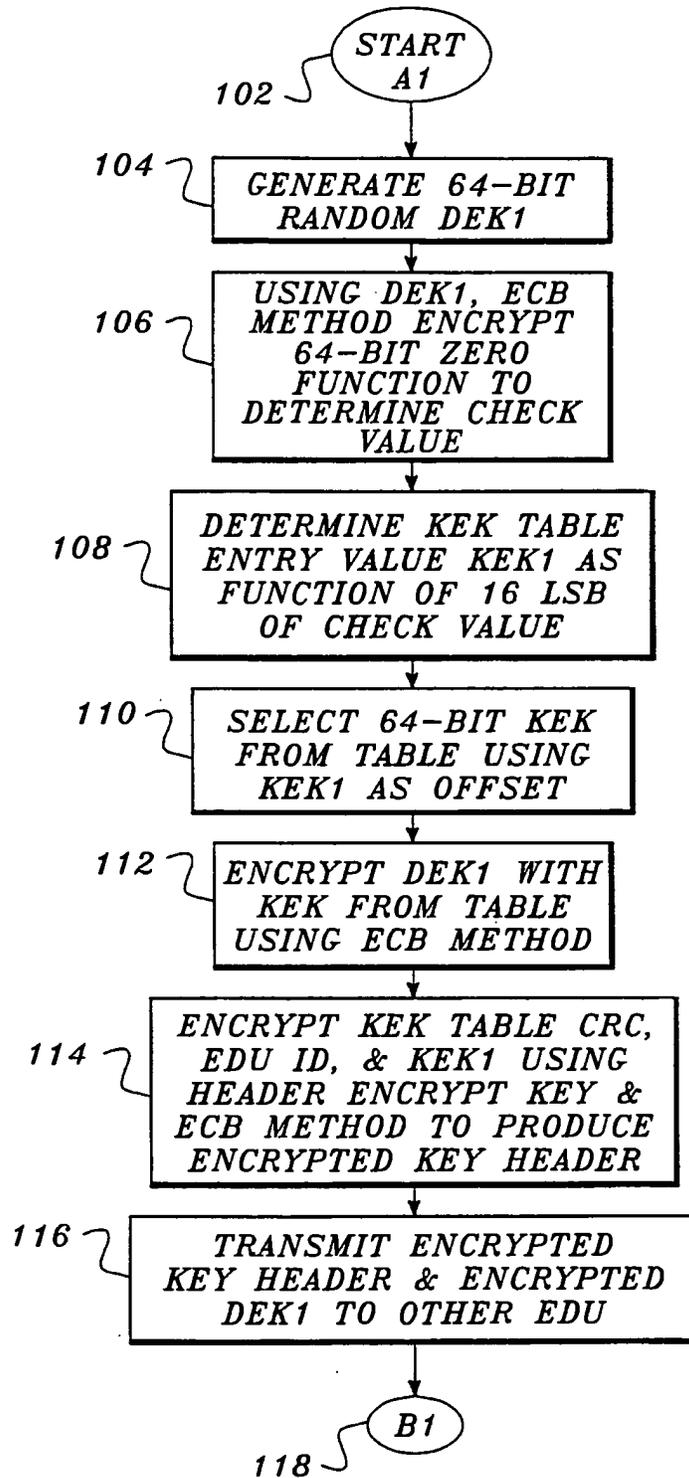


FIG. 2.



100 ↗

FIG. 3.

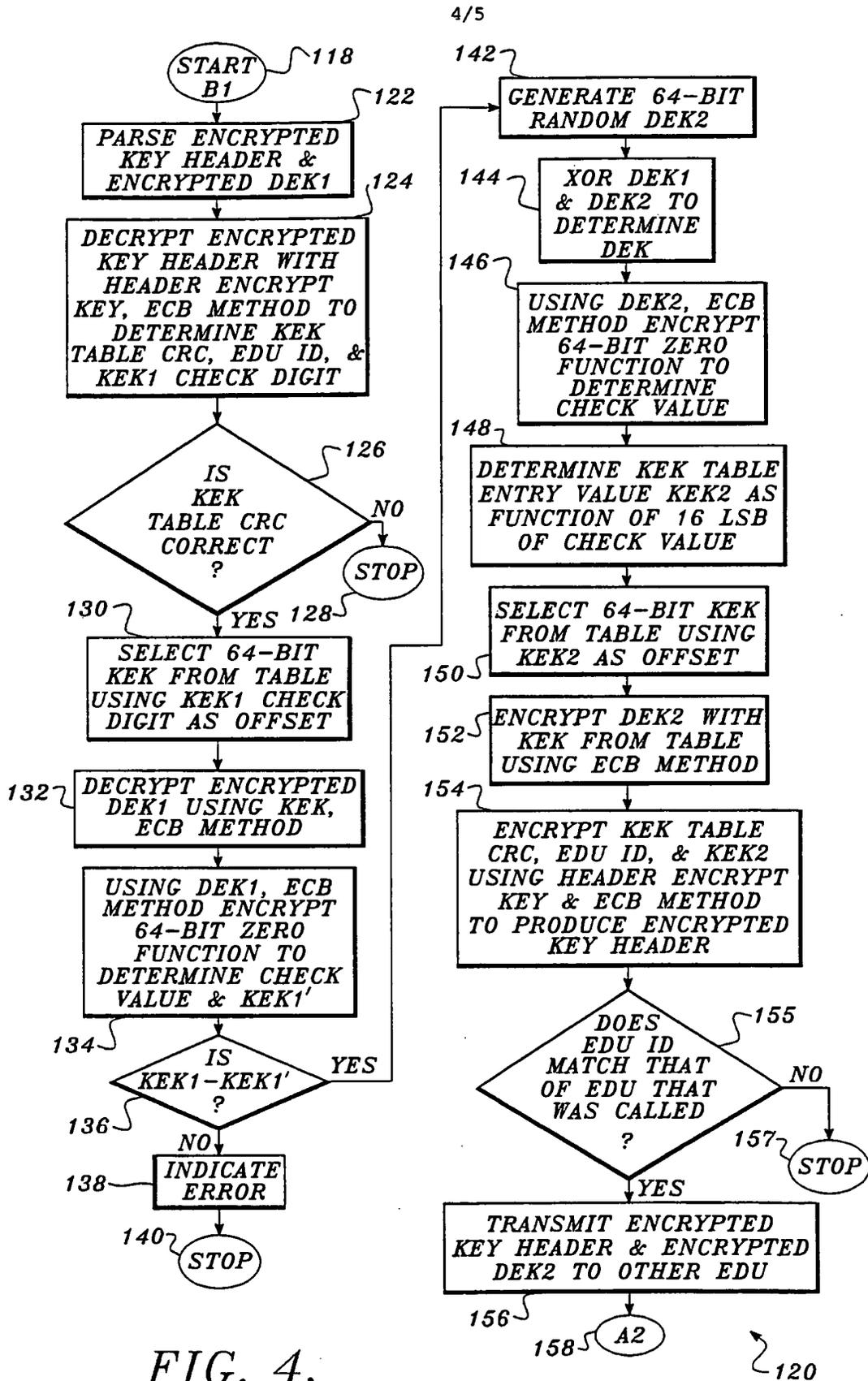


FIG. 4.

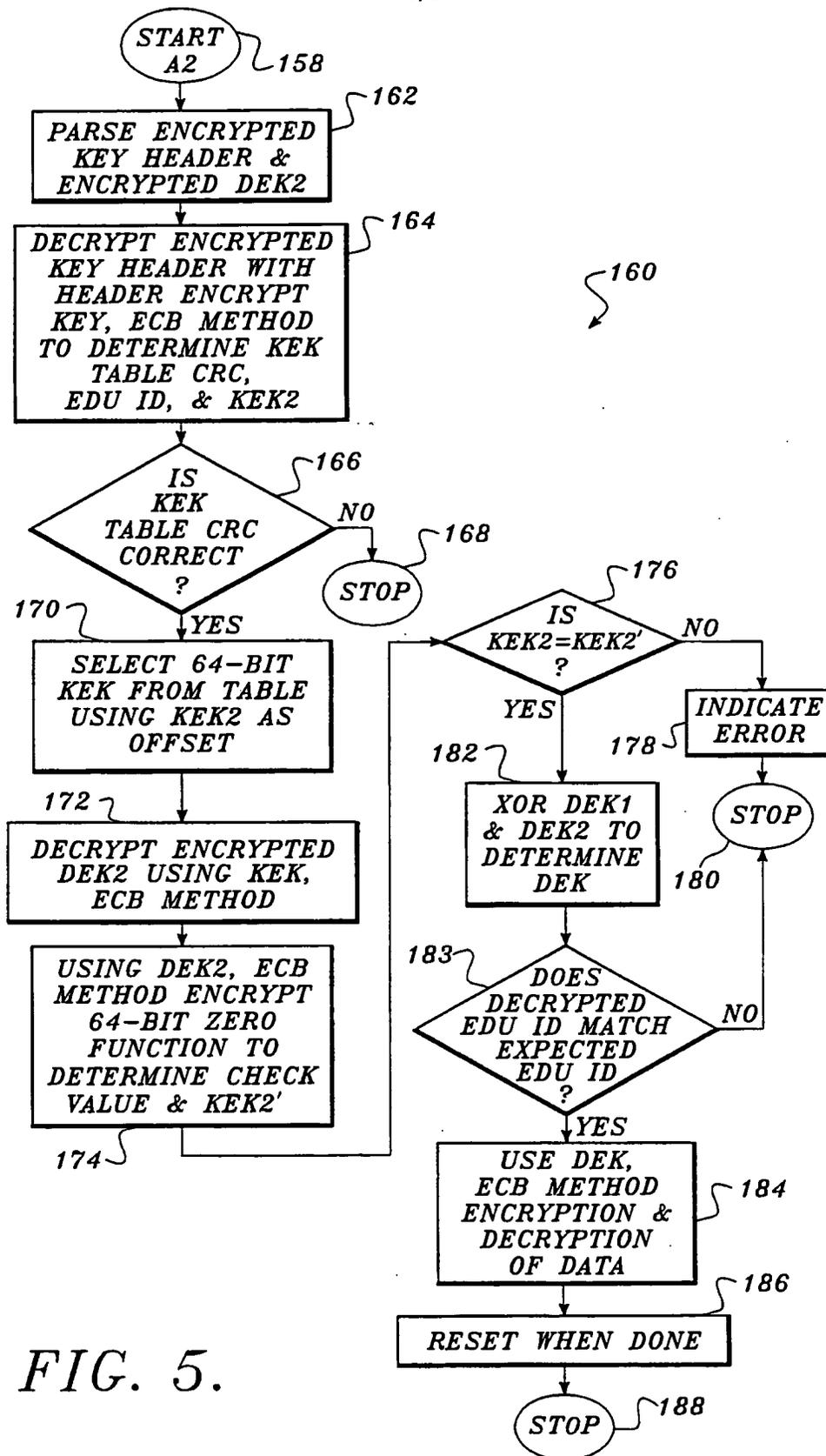
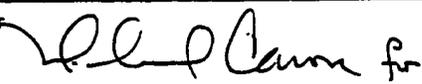


FIG. 5.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US93/04340

A. CLASSIFICATION OF SUBJECT MATTER		
IPC(5) :HO4L 9/00 US CL :380/21, 49, 28 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) U.S. : 380/21, 49, 28		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched 380/52, 46		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US, A 5,029,208 (Tanaka) 02 July 1991	1-29
A	US, A 5,124,117 (Tatebayashi, et al.) 23 June 1992	1-29
A, P	US, A 5,144,665 (Takaragi, et al.) 01 September 1992	1-29
A	US, A 4,607,137 (Jansen, et al.) 19 August 1986	1-29
A	US, A RE33189 (Lee, et al.) 27 March 1990	1-29
A	US, A 4,578,531 (Everhart, et al.) 25 March 1986	1-29
A	US, A 4,876,716 (Okamoto) 24 October 1989	1-29
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* "A"	Special categories of cited documents: document defining the general state of the art which is not considered to be part of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E"	earlier document published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means	"Z" document member of the same patent family
"P"	document published prior to the international filing date but later than the priority date claimed	
Date of the actual completion of the international search 10 August 1993		Date of mailing of the international search report 26 AUG 1993
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. NOT APPLICABLE		Authorized officer David Cain  Telephone No. 308-0463

Form PCT/ISA/210 (second sheet)(July 1992)*



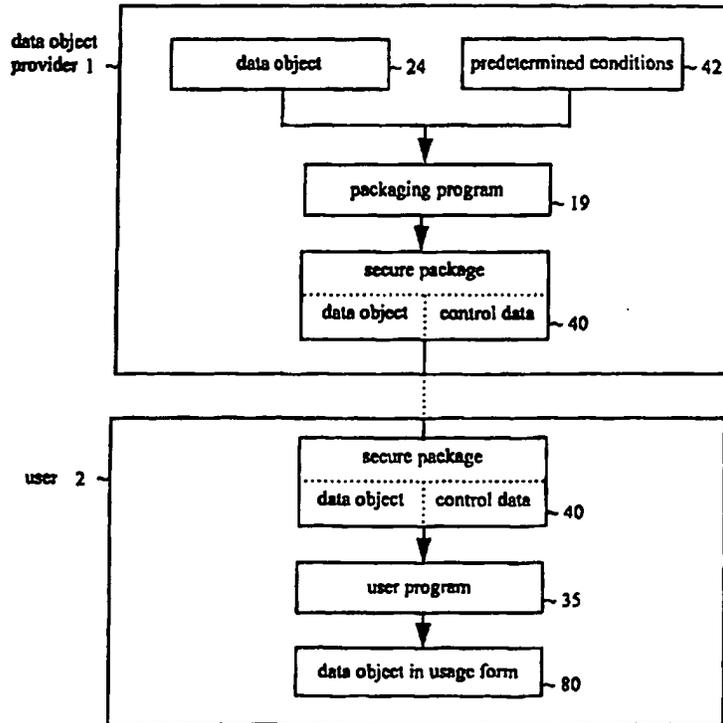
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁶ : G06F 1/00, 12/14</p>	<p>A2</p>	<p>(11) International Publication Number: WO 96/24092 (43) International Publication Date: 8 August 1996 (08.08.96)</p>
<p>(21) International Application Number: PCT/SE96/00115 (22) International Filing Date: 1 February 1996 (01.02.96) (30) Priority Data: 9500355-4 1 February 1995 (01.02.95) SE (71)(72) Applicant and Inventor: BENSON, Greg [US/SE]; Dalbackavägen 3, S-240 10 Dalby (SE). (72) Inventor; and (75) Inventor/Applicant (for US only): URICH, Gregory, H. [US/SE]; Warholmsvägen 8 B, S-224 65 Lund (SE). (74) Agent: AWAPATENT AB; P.O. Box 5117, S-200 71 Malmö (SE).</p>	<p>(81) Designated States: AL, AM, AT, AT (Utility model), AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), TJ, TM, TR, TT, UA, UG, US, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AZ, BY, KG, KZ, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).</p> <p>Published Without international search report and to be republished upon receipt of that report.</p>	

(54) Title: A METHOD AND SYSTEM FOR MANAGING A DATA OBJECT SO AS TO COMPLY WITH PREDETERMINED CONDITIONS FOR USAGE

(57) Abstract

The present invention relates to a method and a system for managing a data object so as to comply with predetermined conditions for usage of the data object. To control the usage of the data object, a set of control data, defining usages of the data object which comply with the predetermined conditions, is created for the data object. The data object is concatenated with the user set of control data, encrypted and transferred to the user. When the user wants to use the data object, a special user program checks whether the usage complies with the control data. If so, the usage is enabled. Otherwise it is disabled.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgystan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Larvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

A METHOD AND SYSTEM FOR MANAGING A DATA OBJECT SO AS TO
COMPLY WITH PREDETERMINED CONDITIONS FOR USAGE

Technical Field

The present invention relates to data processing and more particularly to a method and a system for managing data objects so as to comply with predetermined conditions for usage.

Background

Much has been written recently regarding the puzzle of universal connectivity. A typical vision of the data highway has long distance high speed data carriers interconnecting regional networks which provide telecommunications services and a wide range of interactive on-line services to consumers. Many of the pieces are already in place, others are in development or testing. In fact, even though the data highway is under construction it is currently open to limited traffic. On-line services are springing up daily and video on demand services are currently being tested.

The potential to benefit society is immense. The scope of information available to consumers will become truly global as the traditional barriers to entry for distribution of, and access to, information are lowered dramatically. This means that more diverse and specialized information will be made available just as conveniently as generic sources from major vendors used to be. The end result is that organizations and individuals will be empowered in ways heretofore only imagined.

However, a fully functioning data highway will only be as valuable as the actual services which it provides. Services envisioned for the data highway that involve the delivery of data objects (e.g. books, films, video, news, music, software, games, etc.) will be and are currently limited by the availability of such objects. Library and educational services are similarly affected. Before owners will allow their data objects to be offered they

must be assured of royalty payments and protection from piracy.

Encryption is a key component of any solution to provide copy protection. But encryption alone is not
5 enough. During transmission and storage the data objects will be protected by encryption, but as soon as anyone is given the key to decipher the content he will have unlimited control over it. Since the digital domain permits data objects to be reproduced in unlimited quantities
10 with no loss of quality, each object will need to be protected from unlimited use and unauthorized reproduction and resale.

The protection problem must not be solved by a separate solution for each particular data format, because
15 then the progress will indeed be slow. It is important to consider the effect of standardization on an industry. Consider how the VHS, the CD and the DAT formats, and the IBM PC compatibility standards have encouraged growth in their respective industries. However, if there is to be
20 any type of standardization, the standard must provide universal adaptability to the needs of both data providers and data users.

The data object owner may want to have permanent secure control over how, when, where, and by whom his
25 property is used. Furthermore, he may want to define different rules of engagement for different types of users and different types of security depending on the value of particular objects. The rules defined by him shall govern the automated operations enabled by data
30 services and networking. The owner may also want to sell composite objects with different rules governing each constituent object. Thus, it is necessary to be able to implement variable and extensible control.

The user on his part wants to be able to search for
35 and purchase data objects in a convenient manner. If desired, the user should be able to combine or edit purchased objects (i.e. for creating a presentation).

Furthermore, the user may want to protect his children from inappropriate material. A complete solution must enable these needs as well.

What is needed is a universally adaptable system and
5 method for managing the exchange and usage of data objects while protecting the interests of data object owners and users.

Prior Art

A method for enforcing payment of royalties when
10 copying softcopy books is described in the European patent application EP 0 567 800. This method protects a formatted text stream of a structured document which includes a royalty payment element having a special tag. When the formatted text stream is inputted in the user's
15 data processor, the text stream is searched to identify the royalty payment element and a flag is stored in the memory of the data processor. When the user for instance requests to print the document, the data processor requests authorization for this operation from a second
20 data processor. The second data processor charges the user the amount indicated in the royalty payment element and then transmits the authorization to the first data processor.

One serious limitation of this method is that it can
25 only be applied to structured documents. The description of the above-mentioned European patent application defines a structured document as: a document prepared in accordance with an SGML-compliant type definition. In other words it can not be applied to documents which are
30 not SGML compliant and it cannot be applied to any other types of data objects.

Furthermore, this method does not provide for variable and extensible control. Anyone can purchase a softcopy book on a CD, a floppy disc or the like, and the
35 same royalty amount is indicated in the royalty payment element of all softcopy books of the same title.

Thus, the method described in EP 0 567 800 does not satisfy the above-mentioned requirements for universally adaptable protection of data objects.

Summary of the Invention

5 Accordingly, it is a first object of the invention to provide a method and a data processing system for managing a data object in a manner that is independent of the format and the structure thereof, so as to comply with predetermined conditions for usage control and
10 royalty payment.

 It is a further object of the invention to provide such a method and system which is universally adaptable to the needs of both the owner and the user of the data object.

15 A further object of the invention is to provide such a method and system which enables a data object provider to distribute his data object while maintaining control of the usage thereof.

 Yet another object of the invention is to provide a
20 method and system which allows a data object provider to select the level of security for his data object in a flexible way.

 Yet another object of the invention is to provide such a method and system which makes it possible to
25 establish an audit trail for the data object.

 Yet another object is to provide such a method and system which makes it possible to sell and buy data objects in a secure way.

 The above-mentioned objects are achieved by a method
30 and a system having the features of claims 1, 16, 21, 24 and 27.

 Particular embodiments of the inventions are recited in the subclaims.

 More particularly, a data object provider, e.g. the
35 owner of a data object or his agent (broker), stores the data object in a memory device, e.g. a bulk storage device, where it is accessible by means of the data

provider's data processor. The data object can consist of digital data, analog data or a combination or hybrid of analog and digital data.

5 A general set of control data, which is based on the predetermined conditions for usage of the data object, is created and stored in the same memory device as the data object or another memory device where it is accessible by the data provider's data processor. The predetermined conditions for usage may be defined by the data object
10 owner, by the broker or by anyone else. They may differ between different data objects.

The general set of control data comprises at least one or more usage control elements, which define usages of the data object which comply with the predetermined
15 conditions. These usages may encompass for instance the kind of user, a time limit for usage, a geographical area for usage, allowed operations, such as making a hard copy of the data object or viewing it, and/or claim to royalty payment. The general set of control data may comprise
20 other kinds of control elements besides the usage control element. In a preferred embodiment, the general set of control data comprises a security control element which defines a security procedure which has to be carried out before usage of the data object. It also comprises an
25 identifier, which uniquely identifies the general set of control data.

The general set of control data is concatenated with a copy of the data object. Thus, the control data does not reside in the data object, but outside it, which
30 makes the control data independent of the format of and the kind of data object and which allows for usage control independently of the data object format.

At least the usage control element(s) and the data object are encrypted, so that the user is unable to use
35 the data object without a user program which performs the usage control and which decrypts the data object. Alter-

natively, the whole set of control data and the copy of the data object may be encrypted.

5 A user may request authorization for usage of a data object residing at a data provider's processor via a data network or in any other appropriate way. The authorization may or may not require payment. When a request for authorization for usage is received, a user set of control data is created by the data provider's processor. The user set of control data comprises the general set of control data or a subset thereof including at least one of said usage control elements which is relevant for the actual user. It typically also includes a new identifier which uniquely identifies this set of control data. If relevant, the user set of control data also comprises an indication of the number of usages authorized. If more than one kind of usage is authorized, the number of each kind of usage may be specified. Finally, the user set of control data is concatenated with a copy of the data object, and at least the usage control elements and the copy of the data object are encrypted to create a secure data package ready for transfer to the user.

15 Before the data package is transferred to the user, it should be confirmed that the request for authorization for usage has been granted. The check is preferably carried out before the user set of control data is created. However, it can also be carried out in parallel with or after the creation of the user control data. In the latter case, the number of usages requested by the user is tentatively authorized and included in the user set, but if the request is refused the user set is cancelled or changed.

25 The data package may be transferred to the user by electronic means or stored on bulk storage media and transferred to the user by mail or by any suitable transportation means.

30 Once the data object has been packaged in the above-described manner, it can only be accessed by a user

program which has built-in usage control and means for
decrypting the data package. The user program will only
permit usages defined as acceptable in the control data.
Moreover, if the control data comprises a security con-
5 trol element, the security procedure prescribed therein
has to be complied with. In one embodiment, the usage
control may be performed as follows. If the user decides
to use a data object, the user program checks the control
data to see if this action is authorized. More particu-
10 larly, it checks that the number of authorized usages of
this kind is one or more. If so, the action is enabled
and the number of authorized usages decremented by one.
Otherwise, the action is interrupted by the user program
and the user may or may not be given the opportunity to
15 purchase the right to complete the action.

After the usage, the user program repackages the
data object in the same manner as it was packaged before.

When a data object is redistributed by a user or a
broker, new control elements are added in the control
20 data to reflect the relation between the old user/broker
and the new user/broker. In this way, an audit trail for
the data object may be created.

According to another aspect of the invention at
least two data packages are stored on a user's data
25 processor, which examines the usage control elements of
the data packages in order to find a match. If a match is
found, the user's data processor carries out an action
which is specified in the user set of control data. This
method can be used for selling and buying data objects.

30 Brief Description of Drawings

Fig. 1 is a flow diagram showing the general data
flow according to the invention.

Fig. 2 is a system block diagram of a data object
provider's data processor.

35 Fig. 3 is a block diagram showing the different
modules of a data packaging program according to the
invention.

Fig. 4 is a data flow diagram of a data packaging process.

Fig. 5 is an example of a header file.

Fig. 6 is an example of a usage data file.

5 Fig. 7 is a data flow diagram of loading an object to the data object provider's data processor.

10 Figs 8a and 8b are examples of control data for a data object on the data object provider's data processor and for an object ready to be transferred to a user, respectively.

Fig. 9 is a data flow diagram of data packaging on the data object provider's data processor.

Fig. 10 is a flow diagram of a data packaging procedure.

15 Fig. 11 is a memory image of a data object and its control data.

Fig. 12a is a memory image of the concatenated control data and data object.

20 Fig. 12b is a memory image of the concatenated and encrypted control data and data object.

Fig. 13 is a system block diagram of a user's data processor.

Fig. 14 is a block diagram showing the different modules of a user program according to the invention.

25 Fig. 15 is a flow diagram of using a data object on the user's data processor.

Fig. 16 is a flow diagram of how the user program operates in a specific application example.

30 Fig. 17 is an example of various data package structures for composite objects.

Description of the Best Mode for Carrying Out the Invention

General Overview

35 Fig. 1 is a flow diagram showing the general data flow according to the invention. The flow diagram is divided into a data object provider part 1 and a user part 2.

In the data object provider part 1, a data object 24 is created by an author. The data object can consist of digital data, analog data or a combination or hybrid of analog and digital data. The primary difference between
5 analog data objects and digital data objects is the means for storage, transfer and usage.

The author also determines the conditions 42 for the usage of the data object 24 by a user. The data object 24 and the usage conditions 42 are input to a data packaging
10 program 19, which creates a secure data package 40 of the data object and of control data which are based on the input usage conditions 42. Once packaged in this way, the data object can only be accessed by a user program 35.

The data object may be packaged together with a
15 general set of control data, which is the same for all users of the data object. This may be the case when the data object is sent to a retailer or a bulletin board, wherefrom a user may obtain it. The data object may also be packaged as a consequence of a request from a user for
20 usage of the data object. In that case, the package may include control data which is specifically adapted to that user. This control data is called a user set of control data. It may for example comprise the number of usages purchased by the user. Typically, the user set of
25 control data will be created on the basis of the general set of control data and include at least a subset thereof. A user set of control data need not always be adapted for a specific user. All sets of control data which are created on the basis of a general set of control data
30 will be called a user set of control data. Thus, a set of control data can be a general set in one phase and a user set in another phase.

The above-mentioned data packaging can be carried out by the author himself by means of the data packaging
35 program 19. As an alternative, the author may send his data object to a broker, who inputs the data object and the usage conditions determined by the author to the data

packaging program 19 in order to create a secure package 3. The author may also sell his data object to the broker. In that case, the broker probably wants to apply his own usage conditions to the data packaging program.

5 The author may also provide the data object in a secure package to the broker, who repackages the data object and adds further control data which is relevant to his business activities. Various combinations of the above alternatives are also conceivable.

10 In the user part 2 of the flow diagram, the secure package 40 is received by a user, who must use the user program 35 in order to unpackage the secure package 40 and obtain the data object in a final form 80 for usage. After usage, the data object is repackaged into the
15 secure package 40.

The different parts of the system and the different steps of the method according to the invention will now be described in more detail.

The data provider's data processor:

20 Fig. 2 is a system block diagram of a data object provider's data processor. As mentioned above, the data object provider may be an author of a data object, an owner of a data object, a broker of a data object or anyone else who wants to distribute a data object, while
25 retaining the control of its usage. The data processor is a general or special purpose processor, preferably with network capabilities. It comprises a CPU 10, a memory 11 and a network adapter 12, which are interconnected by a bus 13. As shown in Fig. 2, other conventional means,
30 such as a display 14, a keyboard 15, a printer 16, a bulk storage device 17, and a ROM 18, may also be connected to the bus 13. The memory 11 stores network and telecommunications programs 21 and an operating system (OS) 23. All the above-mentioned elements are well-known to the
35 skilled person and commercially available. For the purpose of the present invention, the memory 11 also stores a data packaging program 19 and, preferably, a database

20 intended for control data. Depending upon the current operation, one or more data objects 24 can be stored in the memory 11 as shown or in the bulk storage 17. The data provider's data processor is considered secure.

5 The Data Packaging Program:

The data packaging program 19 is used for creating control data for controlling the usage of a data object and for packaging the data object and the control data into a secure package.

10 As shown in Fig. 3, it comprises a program control module 301, a user interface module 302, a packaging module 303, a control data creation module 304, an encryption module 305, one or more format modules 306, and one or more security modules 307.

15 The control module 301 controls the execution of the other modules. The user interface module 302 handles interaction with the data object provider. The packaging module 303 packages the control data and the data object. It uses the control data creation module 304, the format
20 modules 306, the security modules 307 and the encryption module 305 as will be described more in detail below.

The format modules 306 comprise program code, which is required to handle the data objects in their native format. They can fulfill functions such as data compression and data conversion. They can be implemented by any
25 appropriate, commercially available program, such as by means of a routine from the PKWARE Inc. Data Compression Library for Windows and the Image Alchemy package from Handmade Software Incorporated, respectively. They can
30 also be implemented by custom designed programs.

The security modules 307 comprise program code required to implement security, such as more sophisticated encryption than what is provided by the encryption module 305, authorization algorithms, access control and usage
35 control, above and beyond the basic security inherent in the data package.

The data packaging program 19 can contain many different types of both format and security modules. The program control module 301 applies the format and security modules which are requested by the data provider.

5 The encryption module 305 may be any appropriate, commercially available module, such as "FileCrypt" Visual Basic subprogram found in Crescent Software's QuickPak Professional for Windows - FILECRPT.BAS, or a custom designed encryption program.

10 The control data creation module 304 creates the control data for controlling the usage of the data object. An example of a control data structure will be described more in detail below.

The Control Data:

15 The control data can be stored in a header file and a usage data file. In a preferred embodiment, the header file comprises fields to store an object identifier, which uniquely identifies the control data and/or its associated data object, a title, a format code, and a
20 security code. The format code may represent the format or position of fields in the usage data file. Alternatively, the format code may designate one or more format modules to be used by the data packaging program or the user program. The security code may represent the en-
25 ryption method used by the encryption module 305 or any security module to be used by the data packaging program and the user program. The header file fields will be referred to as header elements.

30 The usage data file comprises at least one field for storing data which controls usage of the data object. One or more usage data fields which represent one condition for the usage of the data object will be referred to as a usage element. In a preferred embodiment, each usage element is defined by an identifier field, e.g. a serial
35 number, a size field, which specifies the size of the usage element in bytes or in any other appropriate way, and a data field.

The header elements and the usage elements are control elements which control all operations relating to the usage of the object. The number of control elements is unlimited. The data provider may define any number of control elements to represent his predetermined conditions of usage of the data object. The only restriction is that the data packaging program 19 and the user program 35 must have compatible program code to handle all the control elements. This program code resides in the packaging module and the usage manager module, to be described below.

Control elements can contain data, script or program code which is executed by the user program 35 to control usage of the related data object. Script and program code can contain conditional statements and the like which are processed with the relevant object and system parameters on the user's data processor. It would also be possible to use a control element to specify a specific proprietary user program which can only be obtained from a particular broker.

It is evident that the control data structure described above is but one example. The control data structure may be defined in many different ways with different control elements. For example, the partitioning of the control data in header data and usage data is not mandatory. Furthermore, the control elements mentioned above are but examples. The control data format may be unique, e.g. different for different data providers, or defined according to a standard.

30 The operation of the data packaging program

The operation of a first embodiment of the data packaging program will now be described with reference to the block diagram of Fig. 3 and the flow diagram of Fig. 4.

35 First a data provider creates a data object and saves it to a file, step 401. When the data packaging program is started, step 402, the user interface module

302 prompts the data object provider to input, step 403, the header information consisting of e.g. an object identifier, a title of the data object, a format code specifying any format module to be used for converting the
5 format of the data object, and a security code specifying any security module to be used for adding further security to the data object. Furthermore, the user interface module 302 prompts the data object provider to input
10 usage information, e.g. his conditions for the usage of the data object. The usage information may comprise the kind of user who is authorized to use the data object, the price for different usages of the object etc. The header information and the usage information, which may be entered in the form of predetermined codes, is then
15 passed to the control module 301, which calls the packaging module 303 and passes the information to it.

The packaging module 303 calls the control data creation module 304, which first creates a header file, then creates header data on the basis of the header
20 information entered by the data object provider and finally stores the header data, step 404-405. Then a usage data file is created, usage data created on the basis of the usage information entered by the data provider, and finally the usage data is stored in the usage
25 data file, step 406-407.

The packaging module 303 then applies any format and security modules 306, 307 specified in the header file, steps 408-413, to the data object.

Next, the packaging module 303 concatenates the
30 usage data file and the data object and stores the result as a temporary file, step 414. The packaging module 303 calls the encryption module 305, which encrypts the temporary file, step 415. The level of security will depend somewhat on the quality of the encryption and key methods
35 used.

Finally, the packaging module 303 concatenates the header file and the encrypted temporary file and saves

the result as a single file, step 416. This final file is the data package which may now be distributed by file transfer over a network, or on storage media such as CD-ROM or diskette, or by some other means.

5 Example 1

An example of how the data packaging program 19 can be used will now be described with reference to Figs 5 and 6. In this example the data object provider is a computer graphics artist, who wants to distribute an image that can be used as clip art, but only in a document or file which is packaged according to the method of the invention and which has usage conditions which do not permit further cutting or pasting. The artist wants to provide a free preview of the image, but also wants to be paid on a per use basis unless the user is willing to pay a rather substantial fee for unlimited use. The artist will handle payment and usage authorization on a dial-up line to his data processor.

The artist uses some image creation application, such as Adobe's Photoshop to create his image. The artist then saves the image to file in an appropriate format for distribution, such as the Graphical Interchange Format (GIF). The artist then starts his data packaging program and enters an object identifier, a title, a format code and a security code, which in this example are "123456789", "image", "a", and "b", respectively. In this example, the format code "a" indicates that no format code need be applied, and this code is selected since the GIF format is appropriate and already compressed. Furthermore, the security code "b" indicates that no security module need be applied and this code is selected since the security achieved by the encryption performed by means of the encryption module 305 is considered appropriate by the artist.

Then the artist enters his dial-up phone number, his price for a single use of the image and for unlimited use of the data object, a code for usage types approved, and

for number of usages approved. For this purpose, the user interface module 302 may display a data entry form.

5 The data packaging program 19 creates control data on the basis of the information entered by the artist and stores the data in the header file and in the usage data file as shown in Figs 5 and 6, respectively. This data constitutes a general set of control data which is not specifically adapted to a single user, but which indicates the conditions of usage determined by the artist
10 for all future users.

Then the package program 19 concatenates the data object and the control data in accordance with steps 414-416 of Fig. 4 to achieve the secure package. No format module or security module is applied to the data
15 object, since they are not needed according to the data in the header file.

When the secure package has been obtained, the artist sends it to a bulletin board, from where it can be retrieved by a user.

20 Example 2

Below, another embodiment of the data packaging program 19 will be described with reference to Figs 7-12b. In this example, the data object consists of a video film, which is created by a film company and sent to a
25 broker together with the predetermined conditions 42 for usage of the video. The broker loads the video 24 to the bulk storage 17 of his data processor. Then, he uses his data packaging program 19 to create a general set of control data 50 based on the predetermined conditions 42
30 for usage indicated by the film company. Furthermore, the address to the video in the bulk storage 17 is stored in an address table in the control database 20 or somewhere else in the memory 11. It could also be stored in the general set of control data 50. Finally, the general set
35 of control data 50 is stored in the control database 20. It could also be stored somewhere else in the memory 11.

After these operations, which correspond to steps 401-407 of Fig. 4, the data packaging program is exited.

Fig. 8a shows the general set of control data for the video according to this example. Here the control data includes an identifier, a format code, a security code, the number of usage elements, the size of the data object, the size of the usage elements and two usage elements, each comprising an identifier field, a size field and a data field. The identifier may be a unique number in a series registered for the particular broker. In this example, the identifier is "123456789", the format code "0010", which, in this example, indicates the format of a AVI video and the security code is "0010". Furthermore, the first usage element defines the acceptable users for the video and the second usage element data defines the number of viewings of the video purchased by a user. The first usage element data is 1 which, for the purposes of this example will signify that only education oriented users are acceptable to the film company. The data field of the second usage element data is empty, since at this stage no viewings of the video has been purchased.

Managing Object Transfer:

The broker wants to transfer data objects to users and enable controlled usage in return for payment of usage fees or royalties. Managing the broker-user business relationship and negotiating the transaction between the broker and the user can both be automated, and the control data structure can provide unlimited support to these operations. The payment can be handled by transmitting credit card information, or the user can have a debit or credit account with the broker which is password activated. Preferably, payment should be confirmed before the data object is transferred to the user.

Data packaging:

When a user wants to use a data object, he contacts the broker and requests authorization for usage of the data object. When the request for authorization is recei-

ved in the broker's data processor, a data program compares the usage for which authorization is requested with the usage control elements of the control data of the data object to see if it complies with the predetermined
5 conditions for usage indicated therein. The comparison may include comparing the user type, the usage type, the number of usages, the price etc. If the requested usage complies with the predetermined conditions the authorization is granted, otherwise it is rejected.

10 Fig. 9 is a data flow diagram of the data packaging on the broker's data processor, which occurs in response to a granted request from a user for authorization for usage of the video, e.g. a granted request for the purchase of two viewings.

15 In response to a granted request, the broker again applies the data packaging program 19. The general set of control data 50 and the data object 24 are input to the program from the control database 20 and the bulk storage 17, respectively. The program creates a user set of control
20 data 60 on the basis of the general set of control data 50 and concatenates the user set 60 and the data object 24 to create a secure data package 40, which may then be transferred to the user by any suitable means. A copy of the user set of control data is preferably stored
25 in the broker's control database. This gives the broker a record with which to compare subsequent use, e.g. when a dial-up is required for usage.

Fig. 10 is a flow diagram of an exemplary procedure used for creating a user set of control data and for
30 packaging the user set of control data and the video into a secure package. Here, the procedure will be described with reference to the general set of control data shown in Fig. 8a.

35 The user set of control data 60, i.e. a set of control data which is adapted to the specific user of this example, is created in steps 1001-1003 of Fig. 11. First, the general set of control data 50 stored in the control

database is copied to create new control data, step 1001. Second, a new identifier, here "123456790", which uniquely identifies the user set of control data, is stored in the identifier field of the new control data 60, step 5 1002. Third, the data field of the second usage element is updated with the usage purchased, i.e. in this example with two, since two viewings of the video were purchased, step 1003.

The thus-created user set of control data, which 10 corresponds to the general set of control data of Fig. 8a is shown in Fig. 8b.

The user set of control data is stored in the control database 20, step 1004. Then, the video, which is stored in the bulk storage 17, is copied, step 1005. The 15 copy of the video is concatenated with the user set of control data, step 1006. The security code 0010 specifies that the entire data package 40 is to be encrypted and that the user program 35 must contain a key which can be applied. Accordingly, the whole data package is encrypted, 20 step 1007. Finally, the encrypted data package is stored on a storage media or passed to a network program, step 1008, for further transfer to the user.

Fig. 11 is a memory image of the video 24 and the user control data 60. The user control data and a copy of 25 the video 24 are concatenated as shown in Fig. 12a. The encrypted data package 40 is shown in Fig. 12b.

The procedure of Fig. 10 can be implemented by the data packaging program of Fig. 3. As an alternative to the procedure of Fig. 10, the user set of control data 30 can be created as in steps 1001-1003 and saved in a header file and in a usage data file, whereafter steps 408-416 of the data packaging program of Fig. 4 can be performed to create the secure package.

The above-described process for creating a user- 35 adapted set of control data may also be used by a user who wants to redistribute a data object or by a broker who wants to distribute the data object to other brokers.

Obviously, redistribution of the data object requires that redistribution is a usage approved of in the control data of the data object. If so, the user or the broker creates a user set of control data by adding new control elements and possibly changing the data fields of old control element to reflect the relation between the author and the current user/broker and between the current user/broker and the future user/broker. In this way, an audit trail is created.

10 The user's data processor:

The user's data processor, which is shown in Fig. 13, is a general or special purpose processor, preferably with network capabilities. It comprises a CPU 25, a memory 26, and a network adapter 27, which are interconnected by a bus 28. As shown in Fig. 13, other conventional means, such as a display 29, a keyboard 30, a printer 31, a sound system 32, a ROM 33, and a bulk storage device 34, may also be connected to the bus 28. The memory 26 stores network and telecommunications programs 37 and an operating system (OS) 39. All the above-mentioned elements are well-known to the skilled person and commercially available. For the purpose of the present invention, the memory 26 also stores a user program 35 and, preferably, a database 36 intended for the control data. Depending upon the current operation, a data package 40 can be stored in the memory 26, as shown, or in the bulk storage 34.

25 The user program:

The user program 35 controls the usage of a data object in accordance with the control data, which is included in the data package together with the data object.

As shown in Fig. 14, the user program 35 comprises a program control module 1401 a user interface module 1402, a usage manager module 1403, a control data parser module 1404, a decryption module 1405, one or more format modules 1406, one or more security modules 1407, and a file transfer program 1409.

The control module 1401 controls the execution of the other modules. The user interface module 1402 handles interactions with the user. The usage manager module 1403 unpackages the secure package 40. It uses the control
5 data parser module 1404, the decryption module 1405, the format modules 1406, and the security modules 1407.

The format modules 1406 comprise program code, which is necessary to handle the data objects in their native format, such as decompression and data format procedures.
10 The security modules 1407 comprises program code required to implement security above the lowest level, such as access control, usage control and more sophisticated decryption than what is provided by the basic decryption module 1405.

15 The user program 35 can contain many different types of both format and security modules. However, they should be complementary with the format and security modules used in the corresponding data packaging program. The usage manager module 1401 applies the format and security
20 modules which are necessary to use a data object and which are specified in its control data. If the proper format and security modules are not available for a particular data object, the usage manager module 1401 will not permit any usage.

25 The decryption module 1405 can be the above-mentioned FileCrypt Visual Basic subprogram or some other commercially available decryption program. It can also be a custom designed decryption module. The only restriction is that the decryption module used in the user program is
30 complementary with the encryption module of the data packaging program.

The control data parser module 1403 performs the reverse process of the control data creation module 304 in Fig. 3.

35 The user program 35 can have code which controls use of the program by password or by any other suitable method. A password may be added in a password control

element during packaging of the data object. The password is transferred to the user by registered mail or in any other appropriate way. In response to the presence of the password control element in the control data structure, the user program prompts the user to input the password. The input password is compared with the password in the control data, and if they match, the user program continues, otherwise it is disabled.

The user program 35 can also have procedures which alter the behavior of the program (e.g. provide filters for children) according to the control data of the user object 41. It is important to mention that the user program 35 never stores the object in native format in user accessible storage and that during display of the data object the print screen key is trapped.

The file transfer program 1409 can transfer and receive files via network to and from other data processor.

Since the data object is repackaged into the secure package after the usage, the user program should also include program code for repackaging the data object. The program code could be the same as that used in the corresponding data packaging program 19. It could also be a separate program which is called from the user program.

25 Operation of the user program:

The operation of an embodiment of the user program 35 will now be described with reference to the block diagram of Fig. 14 and the flow diagram of Fig. 15.

First the user receives a data package 40 via file transfer over a network, or on a storage media such as CD-ROM or diskette, or by any other appropriate means, step 1501. He then stores the data package as a file on his data processor, step 1502.

When the user wants to use the data object, he starts the user program 35, step 1503. Then he requests usage of the data object, step 1504. The request is received by the user interface module 1402, which noti-

fies the control module 1401 of the usage request. The control module 1401 calls the usage manager module 1403 and passes the usage request.

The usage manager module 1403 reads the format code
5 from the data package to determine the control data format. Then it calls the decryption module 1405 to decrypt and extract the control data from the data package. The usage manager module 1403 applies the decryption module 1405 incrementally to decrypt only the control data.

10 Finally, it stores the control data in memory, step 1505.

The usage manager module 1403 then calls the control data parser module 1404 to extract the data fields from the usage elements.

The usage manager module 1403 then compares the user
15 request for usage with the corresponding control data, steps 1506-1507. If the requested usage is not permitted in the control data, the requested usage is disabled, step 1508. However, if the requested usage is approved of in the control data, the usage manager module 1403 applies
20 any format and security modules 1406, 1407 specified in the header data or usage data, steps 1509-1514, to the data package.

Then the usage manager module 1403 calls the decryption
25 module 1405, which decrypts the object data, step 1515, whereafter the requested usage is enabled, step 1516. In connection with the enabling of the usage, the control data may need to be updated, step 1517. The control data may for instance comprise a data field indicating a limited number of usages. If so, this data field
30 is decremented by one in response to the enabling of the usage. When the user has finished usage of the data object, the user program 35 restores the data package in the secure form by repackaging it, step 1518. More particularly, the data object and the usage elements are
35 reconcatenated and reencrypted. Then the header elements are added and the thus-created package is stored in the user's data processor.

Example 1 contd.

A specific example of how the user program operates will now be described with reference to Figs 6 and 15. The example is a continuation of Example 1 above, where
5 an artist created an image and sent it to a bulletin board.

Assume that a user has found the image at an electronic bulletin board (BBS) and is interested in using it. He then loads the data package 40 containing the image to
10 his data processor and stores it as a file in the bulk storage. The user then executes the user program 35 and requests to preview the image. The user program then performs steps 1505-1507 of the flow diagram in Fig. 15. The request for a preview of the image is compared with the
15 data field of the usage element "code for usage type approved". In this example, the code "9" designates that previews are permitted. Thus, the requested preview is OK. Then, the user program 35 performs step 1509-1515 of Fig. 15. Since the format code "a" and the security code
20 "b" of the header data indicate that neither conversion, nor decompression, nor security treatment is required, the user program only decrypts the object data. The usage manager module 1403 then displays the preview on the user's data processor and passes control back to the user
25 interface 1402.

When the user is finished previewing the image, the user interface module 1402 displays the costs for usage of the image in accordance with the price usage data of the control data ("price for single use" and "price for
30 unlimited use" in Fig. 6) and prompts the user to enter a purchase request. The user decides to buy unlimited use of the image, and the user interface module 1402 inputs purchase information, such as an identification, billing, and address for that request and passes the request to
35 the control module 1401. The control module calls the file transfer program 1409, which dials the artist's dial-up number as indicated in the usage data ("control

element for artist's phone number" in Fig. 6) and transfers the request and purchase information to a broker program on the artist's data processor. Upon approval of the purchase, the broker program returns a file containing an update for "usage type approved" control elements. The update is "10" for the usage type approved, which in this example indicates that unlimited use by that user is permitted. The file transfer program 1409 passes this update to the usage manager module 1403 which updates the control data with the "usage type approved" code. The user interface module 1402 then displays a confirmation message to the user.

Subsequently, the user interface module inputs a request to copy the image to a file packaged according to this invention, on the user's machine. The usage manager module then compares the user request control data. The usage manager module examines the data filed for "usage type approved", which now is "10". The usage manager module copies the image to the file.

When the user is finished with the image, the usage manager module 1403 repackages the image as before except with updated control data. This repackaging process is exactly like that shown in Fig. 4, except that the header and usage data already exist, so the process starts after step 406 where control data is created.

Improved security

If the data object provider wants to improve the security of a data package containing a data object, a security module 307 containing a sophisticated encryption algorithm, such as RSA, could be used. In that case the packaging module 303 calls the security module 307 in step 412 of the flow diagram of Fig. 4. The security module encrypts the image and passes a security algorithm code to the control data creation module 302, which adds a control element for the security module code, which will be detected by the user program 35. Then the data packaging continues with step 414. When the data package

is sent to the user, the public key is mailed to the user by registered mail. When the user program is executed in response to a request for usage of this data object, the usage manager module will detect the security module code
5 in the control data and call the security module. This module passes control to the user interface module 1402, which requests the user to input the public key. If the key is correct, the user security module applies complementary decryption using that key and passes a usage
10 approved message to the usage manager module, which enables the usage.

As another example of improved security, a security module may implement an authorization process, according to which each usage of the data object requires a dial-up
15 to the data processor of the data object provider. When the corresponding security module code is detected by the user program 35, the relevant security module is called. This module passes a request for authorization to the control module 1401, which calls the file transfer pro-
20 gram 1409, which dial the data object provider's dial-up number, which is indicated in a usage element and transfers the request for authorization of usage. Upon a granted authorization, the data provider's data processor returns a usage approved message to the user security
25 module, which forwards the approval to the usage control module, which enables one usage. If the user requests further usages of the data object, the authorization process is repeated. This procedure results in a permanent data object security.

30 Example 2 contd.

A further specific example of how the user program 35 operates will now be described with reference to Fig. 16. The example is a continuation of Example 2 above, where a user purchased two viewings of a video film from
35 a broker.

The user wants to play the video which was purchased and transferred from the broker. The user applies the

user program 35, step 1601, and requests to play the video, step 1602. The user program 35 first examines the user set of control data 60, step 1603. In this example, the user program 35 contains only those format and security modules for objects with format code of 0010 and
5 with a security code of 0010. Consequently, only those types of data objects may be used. If the program encounters other codes it will not enable the usage action, step 1604-1605.

10 Next, the user program 35 compares the first control element data which is 1, for educational users only, to user information entered by the user on request of the user program. Since the user type entered by the user is the same as that indicated in the first usage element the
15 process continues, steps 1606-1607. Then the user program checks the second control element data which specifies that the number of plays purchased is 2. Consequently, the usage is enabled, step 1609. The user program applies the decryption module with the universal key and the AVI
20 format video is displayed on the display unit 29. Then, the second control element data is decremented by one, step 1610. Finally, the video is repackaged, step 1611
Implementation of Variable and Extensible Object Control:

Object control is achieved through the interaction
25 of the data packaging program 19 and the usage program 35 with the control data. Variation of object control can be applied to a particular object by creating a control data format with control elements defining the control variation and the circumstances in which the variation is applied.
30 Program procedures should then be added to program modules to process the control elements. For example, suppose a broker wants to allow students to print a particular article for free but require business users to pay for it. He defines control elements to represent the
35 user types student and business and the associated costs for each. He then adds program logic to examine the user type and calculate costs accordingly. Object control is

extensible in the sense that the control data format can have as many elements as there are parameters defining the rules for object control.

Implementation of Variable and Extensible Object

5 Security:

Object security is also achieved through the interaction of the data packaging program 19 and the user program 35 with the control data. Security process and encryption/decryption algorithms can be added as program modules. Variation of object security can be applied to a particular object by creating a control data format with control elements defining the security variation and the circumstances in which the variation is applied. Program procedures should be added to program modules to process the control elements. For example, suppose a broker wants to apply minimal security to his collection of current news articles but to apply tight security to his encyclopedia and text books. He defines a control element for security type. He then adds program logic to apply the security algorithms accordingly. Object security is extensible in the sense that multiple levels of security can be applied. The level of security will of course be dependent on the encryption/key method which is implemented in the security modules. One level of security may be to require online confirmation when loading a data object to the user's data processor. This can be implemented in program code in a security module. This permits the broker to check that the object has not already been loaded as well as double check all other parameters.

30 It is also important to have version control with time stamping between the usage program and the user's control database. Otherwise the database can be duplicated and reapplied to the user program. The user program can place a time stamp in the control database and in a hidden system file each time the control database is accessed. If the time stamps are not identical, the control database has been tampered with and all usage is

disabled. Program code for handling time stamps can reside in a security module.

Handling Composite Objects:

5 A composite object can be handled by defining a control data format with control elements defining relationships between constituent objects and by defining a parent/child element and a related object id element. For example, suppose a broker wants to include a video and a text book in an educational package. He creates a parent
10 object with control elements referring to the video and textbook objects. He also includes control elements in the control data for the video object and the textbook object referring to the parent object. Finally, he adds program procedures to program modules to process the
15 control elements.

In other words, when the data object is a composite data object including at least two constituent data objects, a respective general set of control data is created for each of the constituent data object and the
20 composite data object. In response to a request from a user, a respective user set of control data is created for each of the constituent data objects as well as for the composite data object.

25 Examples of various data package structures for composite objects are given in Fig. 17.

Another side of composite objects is when the user wants to combine data objects for some particular use. Combination is a usage action that must be permitted in each constituent data object. A new data object is
30 created with control data linking the constituent data objects. Each constituent data object retains its original control data which continues to control its subsequent usage.

When a user requests authorization for usage of one
35 constituent data object in a composite data object, a user set of control data is created only for that consti-

tuent data object and concatenated only with a copy of that constituent data object.

Scaleable Implementation:

The flexible control data structure and modular program structure permit almost boundless extensibility with regard to implementation of the owner's requirements for usage control and royalty payment. The control data structure can include control elements for complex user types, usage types, multiple billing schemes, artistic or ownership credit requirements and others. Security modules can be included which interact with any variation of the control data structure and the control data. Security modules could require a dial up to the brokers data processor to approve loading or usage actions and to implement approval authentication mechanisms.

User acting as a broker:

A limited or full implementation of the broker's data packaging program can be implemented on the user's machine to permit further distribution or reselling. However, only those data objects with control data permitting further distribution or reselling are enabled in that way.

Rebrokering

An author of a data object may want to allow his original broker to distribute his data object to other brokers whom will also distribute his image. He then includes a control element which enables rebrokering in the control data before distributing the data object with its associated control data to the original broker. Upon request for rebrokering, the original broker copies the general set of control data and updates the copy to create a user set of control data which will function as the general set of control data on the subsequent brokers data processor. The original broker packages the data object with the user set of control data and transfers the package to the subsequent broker. The subsequent broker then proceeds as if he were an original broker.

Automated transaction negotiation

This is an example of how the predetermined conditions for usage included in the control data can be used for achieving automated transaction negotiation.

5 Suppose some company wants to provide a computer automated stock trading. Buy and sell orders could be implemented in the form of data packages and a user program could process the data packages and execute transactions. Data packages could carry digital cash and
10 manage payment based on conditions defined in the control data.

In this example, the buy order is created using a data packaging program according to the invention on the buyer's data processor. The sell order is created using
15 the data packaging program on the seller's data processor. Both orders are used by the the user program on the stock trader's data processor. The usages would take the form of using a sell order data package to sell stock and a buy order data package to buy stock. The rules or conditions for buying and selling stocks could be indicated
20 in the control data of the packages. The data object consists of digital money. In this context it is important to remember that digital money is merely data which references real money or virtual money that is issued and
25 maintained for the purpose of digital transactions.

In this example the buyer starts with a digital money data file. He uses the data packaging program to create control data, e.g. kind of stock, price, quantity, for the purchase, and he then packages the digital money
30 data file and the control data into a secure package as described above.

The seller starts with an empty data file. This empty file is analogous to the digital money data file except it is empty. The seller creates control data, e.g.
35 kind of stock, price, quantity, and packages the empty file and the control data into a secure package.

Both the sell order package and the buy order package are transferred to the data processor of the stock trading company, where they are received and stored in the memory. The user program of the stock trading company
5 examines the control data of the buy and sell order packages in the same way as has been described above and looks for a match. Upon identifying matched buy and sell orders the user program executes a transaction, whereby the digital money is extracted from the buy order data
10 package and transferred to the sell order package. Then the control data of the data packages is updated to provide an audit trail. Both packages are repackaged in the same manner as they were previously packaged and then transferred back to their authors.

15 The above described technique could be used for selling and buying any object as well as for automated negotiations. Payment may be carried out in other ways than by digital money.

20 In the general case, the data processor of the user decrypts the usage control elements of the user sets of control data and examines the usage control elements to find a match. In response to the finding of a match, the user's data processor carries out an action which is specified in the user set of control data.

25

CLAIMS

1. A method for managing a data object so as to comply with predetermined conditions for usage of the data object, comprising the steps of:
- 5 - storing the data object in a memory device, where it is accessible by means of a data object provider's data processor;
 - creating, by said data processor, a general set of control data for the data object based on said predetermined conditions for usage, said general set of control data comprising at least one or more usage control elements defining usages of the data object which comply with said predetermined conditions;
 - 10 - storing said general set of control data in a memory device, where it is accessible by said data processor;
 - concatenating the general set of control data with a copy of the data object; and
 - 20 - encrypting at least the copy of the data object and said one or more usage control elements to create a secure data package which is ready for transfer to a user.
2. A method as set forth in claim 1, wherein the step of encrypting comprises encrypting the data object and the general set of control data.
3. A method as set forth in claims 1 or 2, wherein the step of creating control data comprises creating an identifier which uniquely identifies the general set of control data.
- 30 4. A method as set forth in claims 1, 2 or 3, wherein the step of creating a general set of control data comprises creating a security control element which identifies a security process to be applied before usage of the data object is allowed.
- 35 5. A method as set forth in any of the preceding claims, wherein the step of creating a general set of

control data comprises creating a format control element which identifies the format of the control data.

6. A method as set forth in any of the preceding claims, comprising the further steps of:

- 5 - creating, in response to a request for authorization for usage of the data object by a user, a user set of control data, which comprises at least a subset of the general set of control data, including at least one of said usage control elements;
- 10 - using the user set of control data instead of the general set of control data in said concatenating step;
- using the at least one usage control element of the user set of control data instead of the one or more usage control elements of the general set of control data
- 15 in the encrypting step;
- checking, before allowing transfer of the data package to the user, that said request for authorization for usage of the data object has been granted.

7. A method as set forth in any of the preceding

20 claims, further comprising the steps of receiving in said data processor the request for authorization for usage by a user; comparing the usage for which authorization is requested with said one or more usage control elements of the general set of control data and granting the authori-

25 zation if the usage for which authorization is requested complies with the usages defined by said one or more usage control elements.

8. A method as set forth in claim 7, further comprising the step of securing payment for the requested

30 authorization for usage before granting the authorization.

9. A method as set forth in any one of claims 6-8, wherein the data object is composed of at least two constituent data objects and wherein the user set of control

35 data, in response to a request for authorization for usage of one of said constituent data objects by a user, is created only for that constituent data object and

concatenated only with a copy of that constituent data object.

10. A method as set forth in any one of claims 6-9, wherein the data provider's data processor is connected
5 to a data network and the request for authorization is received from a data processor of the user, which is also connected to the data network, further comprising the step of transferring the data package through the data network to the user's data processor.

10 11. A method as set forth in any one of claims 6-8 or 10, wherein the data object is a composite data object including at least two constituent data objects and wherein the step of creating a general set of control data comprises the step of creating a respective general
15 set of control data for each of the constituent data objects and the composite data object and wherein the step of creating a user set of control data comprises the step of creating a respective user set of control data for each of the constituent data objects and the compo-
20 site data object.

12. A method as set forth in any one of claims 6-11, comprising the further step of storing a copy of the user set of control data in the data object provider's processor.

25 13. A method as set forth in any of the preceding claims, comprising the further steps of:

- receiving the data package in a user's data processor;
- storing the data package in a memory device where
30 it is accessible by means of the user's data processor;
- decrypting said one or more usage control elements;
- checking, in response to a request by the user for usage of the data object, whether the requested usage
35 complies with the usage defined by the at least one usage control element of the general set of control data;

- decrypting, in response to the requested usage complying with the usage defined by the at least one usage control element of the general set of control data, the data object and enabling the requested usage, otherwise disabling it.

14. A method as set forth in any one of claims 6-12, comprising the further steps of:

- receiving the data package in a user's data processor;

10 - storing the data package in a memory device where it is accessible by means of the user's data processor;

- decrypting the at least one usage control element of the user set of control data;

15 - checking, in response to a request by the user for usage of the data object, whether the requested usage complies with the usage defined by the at least one usage control element of the user set of control data;

20 - decrypting, in response to the requested usage complying with the usage defined by the at least one usage control element of the user set of control data, the data object and enabling the requested usage, otherwise disabling it.

15. A method as set forth in claims 13 or 14, comprising the further steps of reconcatenating, after the usage of the data object, the data object and the one or more usage control elements, reencrypting at least the data object and the one or more usage control elements, and storing the thus-repackaged data package in the memory of the user's data processor.

30 16. A method for controlling the usage by a user of a data object so as to comply with predetermined conditions for usage of the data object, comprising the steps of:

35 - storing a data package in a memory device, where it is accessible by means of a data processor of the user, said data package comprising the data object and control data, which comprises at least one usage control

element defining a usage of the data object which complies with the predetermined conditions, the data object and said at least one usage control element being encrypted;

- 5 - receiving a request by the user for usage of the data object;
- decrypting the control data;
- checking, in response to the request by the user for usage of the data object, whether the requested usage
- 10 complies with the usage defined by the at least one usage control element of the control data;
- decrypting, in response to the requested usage complying with the usage defined by the at least one usage control element of the control data, the data
- 15 object and enabling the requested usage, otherwise disabling it.

17. A method as set forth in claim 16, wherein the usage control element is updated after the usage of the data object.

- 20 18. A method as set forth in claims 16 or 17, wherein said control data comprises an indication of the number of times the user is authorized to use the data object in accordance with said at least one user control element; wherein the requested usage of the data object
- 25 is only enabled when said number of times is one or more; and wherein said number of times is decremented by one when the requested usage is enabled.

19. A method as set forth in any one of claims 16-18, wherein the control data comprise a security control element, and further comprising the step of carrying out, before each usage of the data object, a security procedure defined in the security control element.
- 30

20. A method as set forth in any one of claims 16-19, wherein the step of checking whether the requested
- 35 usage complies with the usage defined by the at least one usage control element comprises the step of checking that the user's data processor is capable of carrying out the

security procedure specified in the security control element of the user set of control data, and if not, disabling the usage.

21. A method as set forth in any one of claims
5 16-20, comprising the further steps of reconcatenating, after the usage of the data object, the data object and the one or more usage control elements, reencrypting at least the data object and the one or more usage control elements, and storing the thus-repackaged data package in
10 the memory of the user's data processor.

22. A system for managing a data object so as to comply with predetermined conditions for usage of the data object, comprising

- first means in the data object provider's data
15 processor for creating a general set of control data for the data object based on the predetermined conditions for usage, said general set of control data comprising at least one or more usage control elements defining usages of the data object which comply with the predetermined
20 conditions;

- storing means, which are accessible by means of said data processor, for storing the data object and the general set of control data;

- concatenating means for concatenating the general
25 set of control data with a copy of the data object; and

- encrypting means for encrypting the copy of the data object and at least said one or more usage control elements to create a secure data package, which is ready for transfer to a user.

30 23. A system as set forth in claim 22, further comprising

- second means in said data processor for creating, in response to a request for authorization for usage of the data object by a user, a user set of control data,
35 which comprises at least a subset of the general set of control data, which subset comprises at least one of said usage control elements; and

- checking means in said data processor for checking that said request for authorization for usage of the data object has been granted before allowing transfer of the data package to the user.

5 24. A system as set forth in claims 22 or 23, wherein the general set of control data comprises a control data element which defines the right to further distribution of the data object by the user.

10 25. A system for controlling the usage by a user of a data object so as to comply with predetermined conditions for usage of the data object, comprising

15 - storing means for storing a data package which comprises a data object and a control data comprising at least one usage control element defining a usage of the data object which complies with the predetermined conditions;

 - means for decrypting the at least one usage control element and the data object;

20 - checking means for checking whether a usage requested by the user complies with the usage defined by said at least one usage control element;

 - enabling means for enabling the usage requested by the user when the usage complies with the usage defined by said at least one usage control element; and

25 - disabling means for disabling the usage requested by the user when the usage does not comply with the usage defined by said at least one usage control element.

30 26. A system as set forth in claim 25, further comprising means for repackaging the data object after usage thereof.

 27. A method for controlling the usage by a user of data objects so as to comply with predetermined conditions for usage of the data objects, comprising the steps of:

35 - storing at least two data packages in a memory device, where they are accessible by a data processor of the user, each said data package comprising a data object

and a user set of control data, which comprises at least one usage control element defining a usage of the data object which complies with the predetermined conditions, the data object and said at least one usage control
5 elements being encrypted;

- decrypting the usage control elements of the user sets of control data;

- examining the usage control elements of said at least two data packages to find a match;

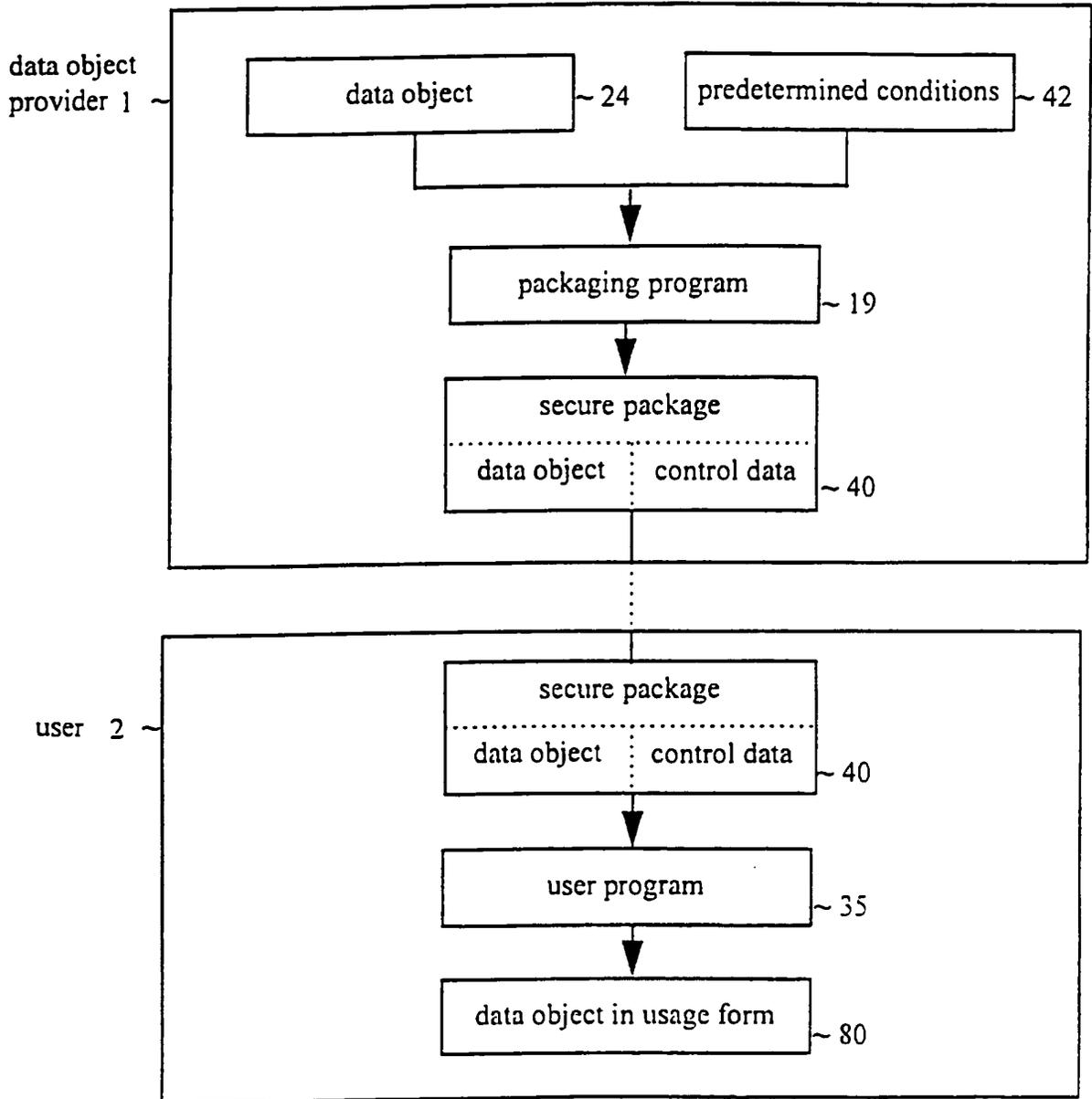
10 - using, in response to the finding of a match, the data processor to carry out an action, which is specified in the user sets of control data.

28. A method as set forth in claim 27, comprising the further steps of updating the usage control element
15 of each data package, reconcatenating after the usage of the data objects, each of the data object and its usage control element, reencrypting each of the concatenated data objects and its usage control element and transferring the repackaged data objects to their creators.

20

1/15

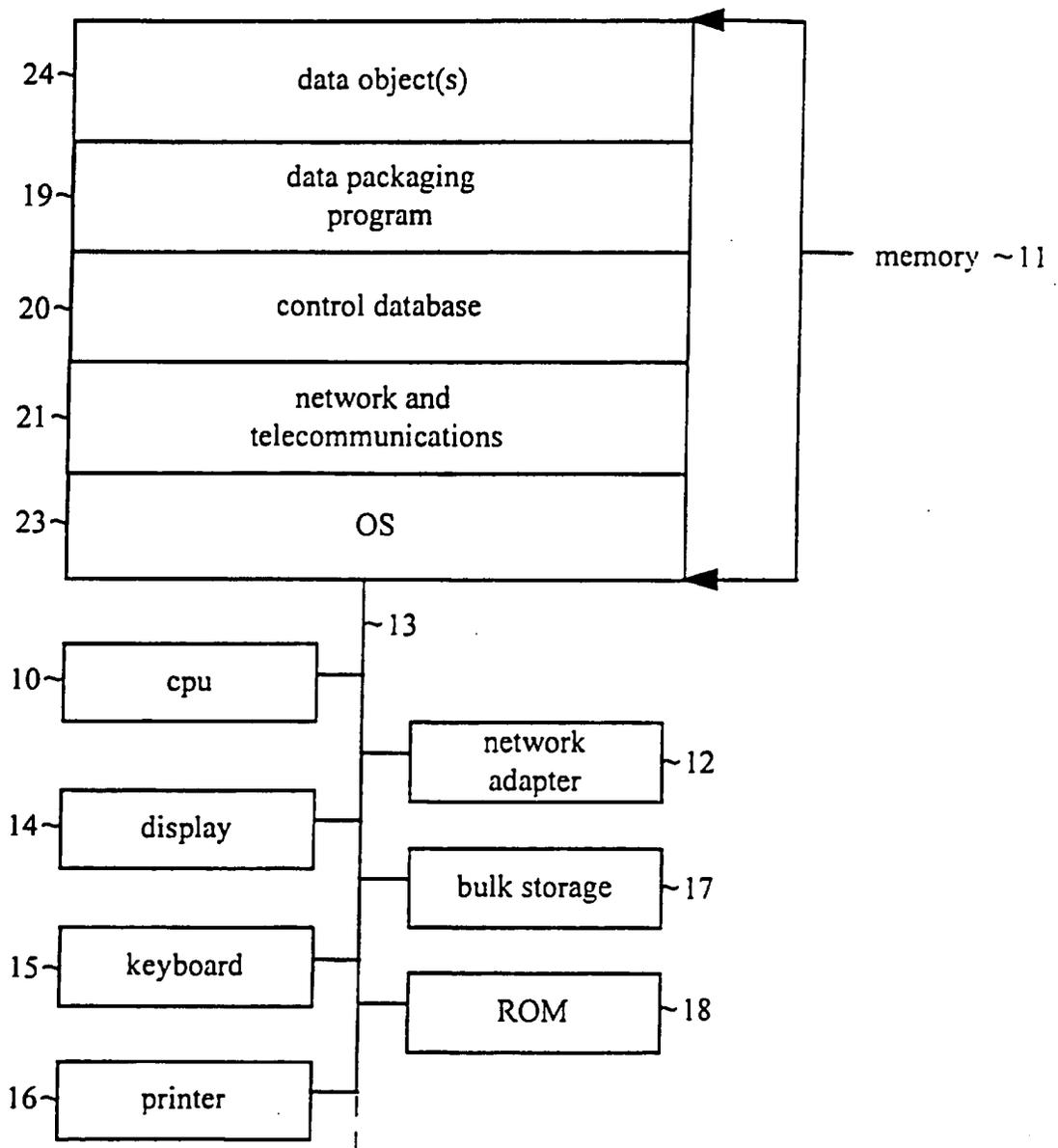
Fig 1



SUBSTITUTE SHEET

2/15

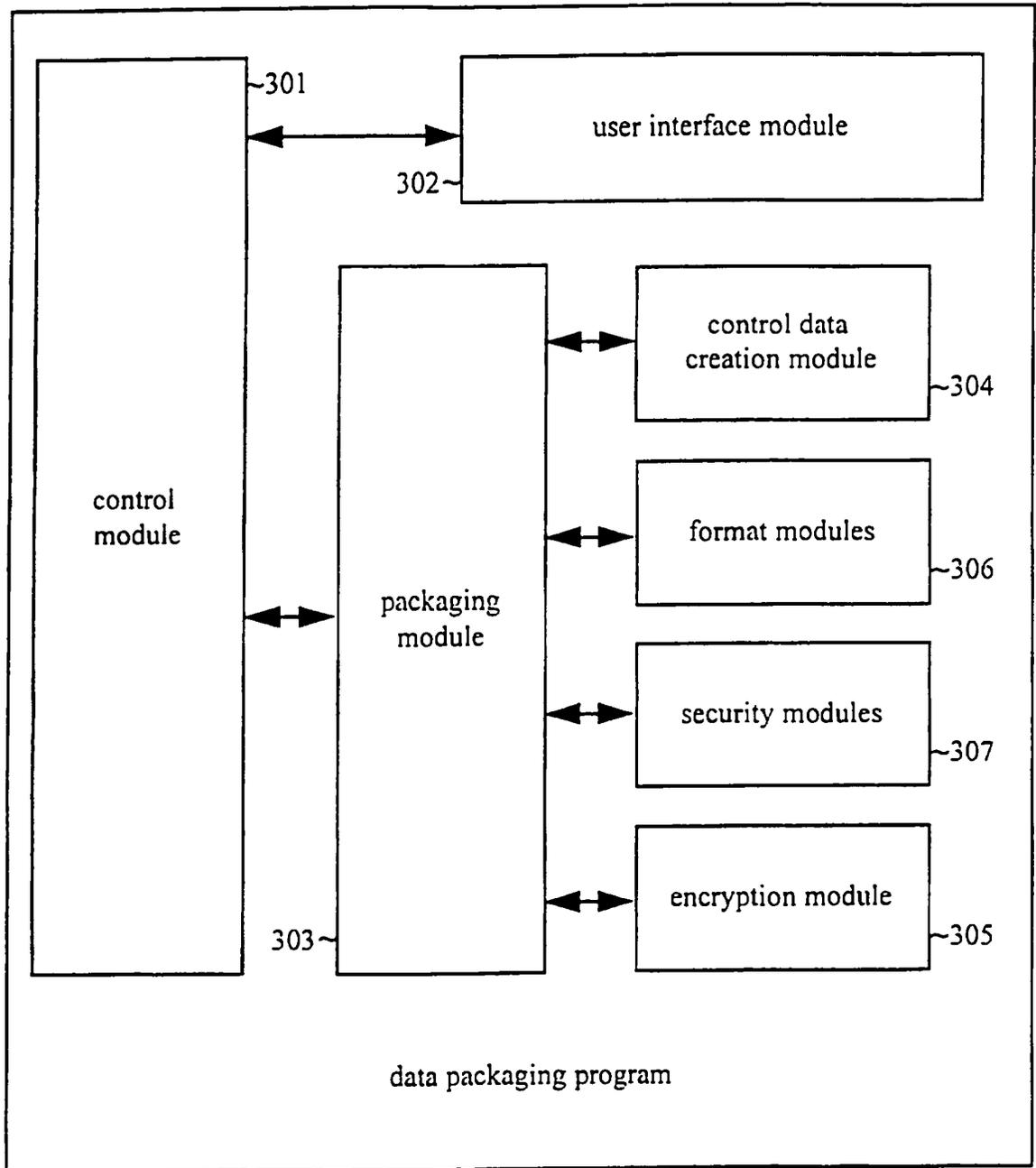
Fig 2



SUBSTITUTE SHEET

3/15

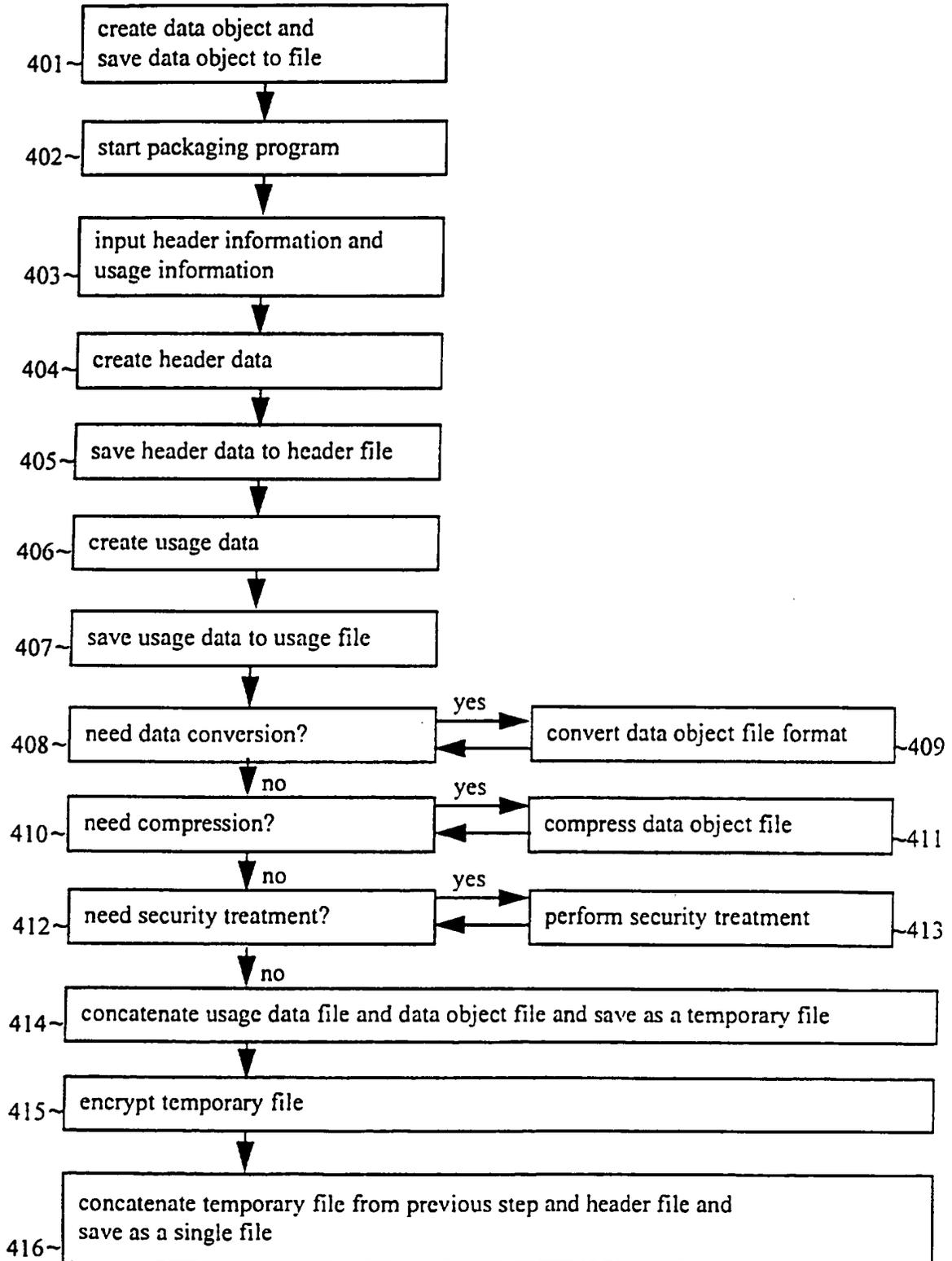
Fig 3



19

4/15

Fig 4



5/15

Fig 5

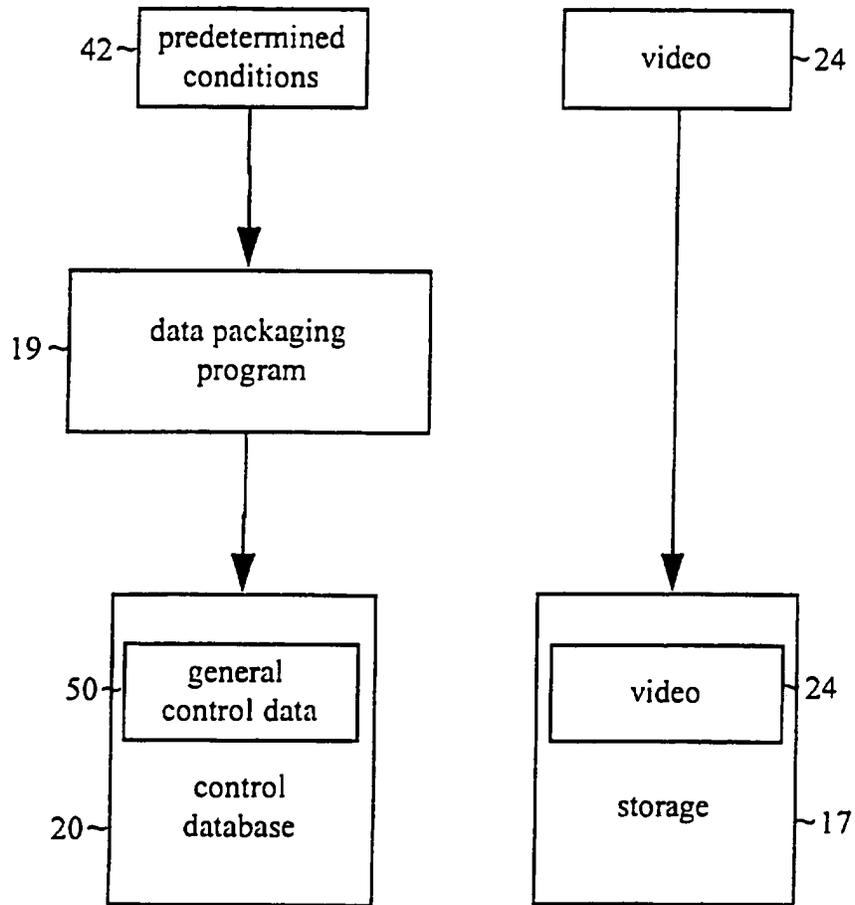
file identifier	123456789
title	image
format code	a
security code	b

Fig 6

usage element for author's phone number	identifer	1
	size	13
	data	716 381 5356
...price for single use	identifer	2
	size	4
	data	.50
...price for unlimited use	identifer	3
	size	4
	data	50.00
...code for usage type approved	identifer	4
	size	2
	data	9
...code for number of usages approved	identifer	5
	size	2
	data	1

6/15

Fig 7



SUBSTITUTE SHEET

7/15
Fig 8a

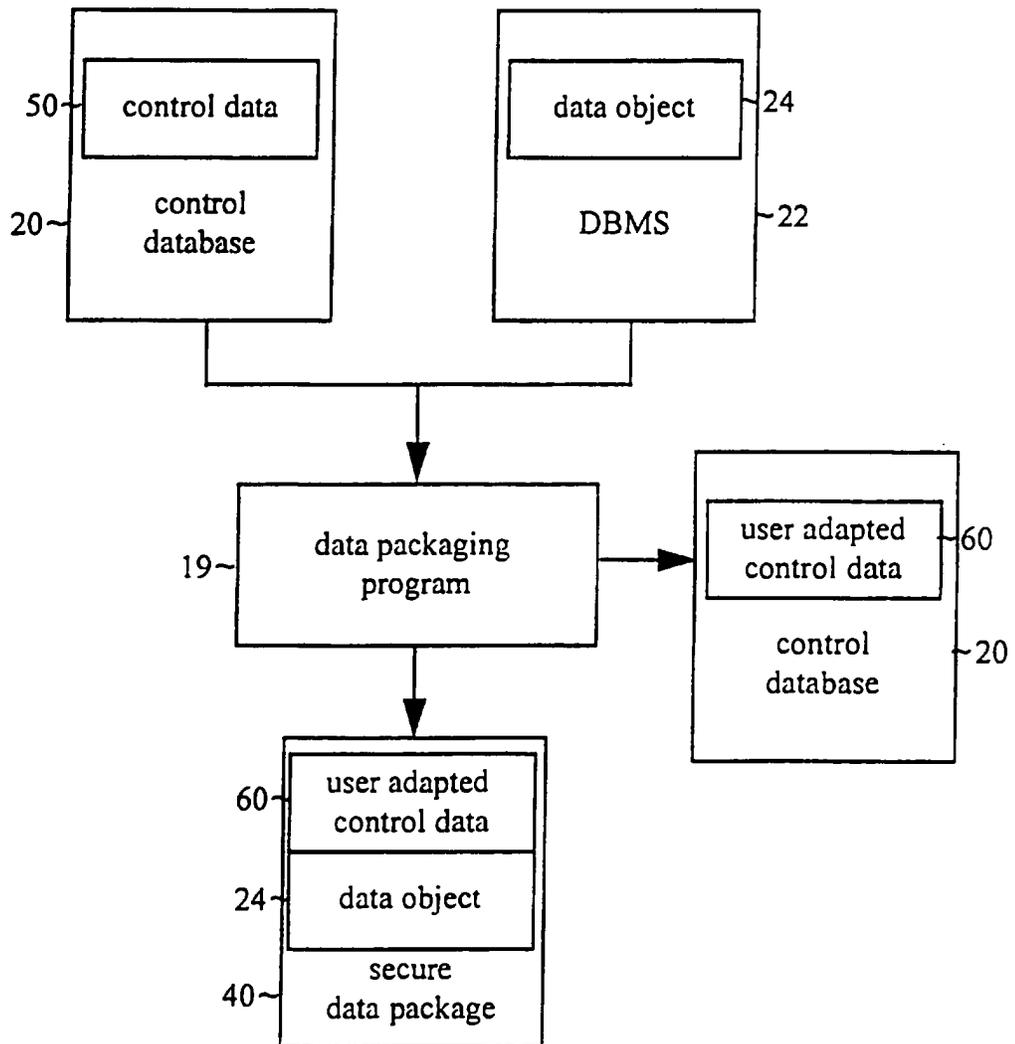
header	object identifier	123456789
	format code	0010
	security code	0010
	number of usage elements	2
	size of usage data	17
	size of data object	273
	1st usage element id	001
	1st usage element size	6
	1st usage element data	1
	2nd usage element id	002
	2nd usage element size	3
	2nd usage element data	

Fig 8b

header	object identifier	123456790
	format code	0010
	security code	0010
	number of usage elements	2
	size of usage data	17
	size of data object	273
	1st usage element id	001
	1st usage element size	6
	1st usage element data	1
	2nd usage element id	002
	2nd usage element size	3
	2nd usage element data	2

8/15

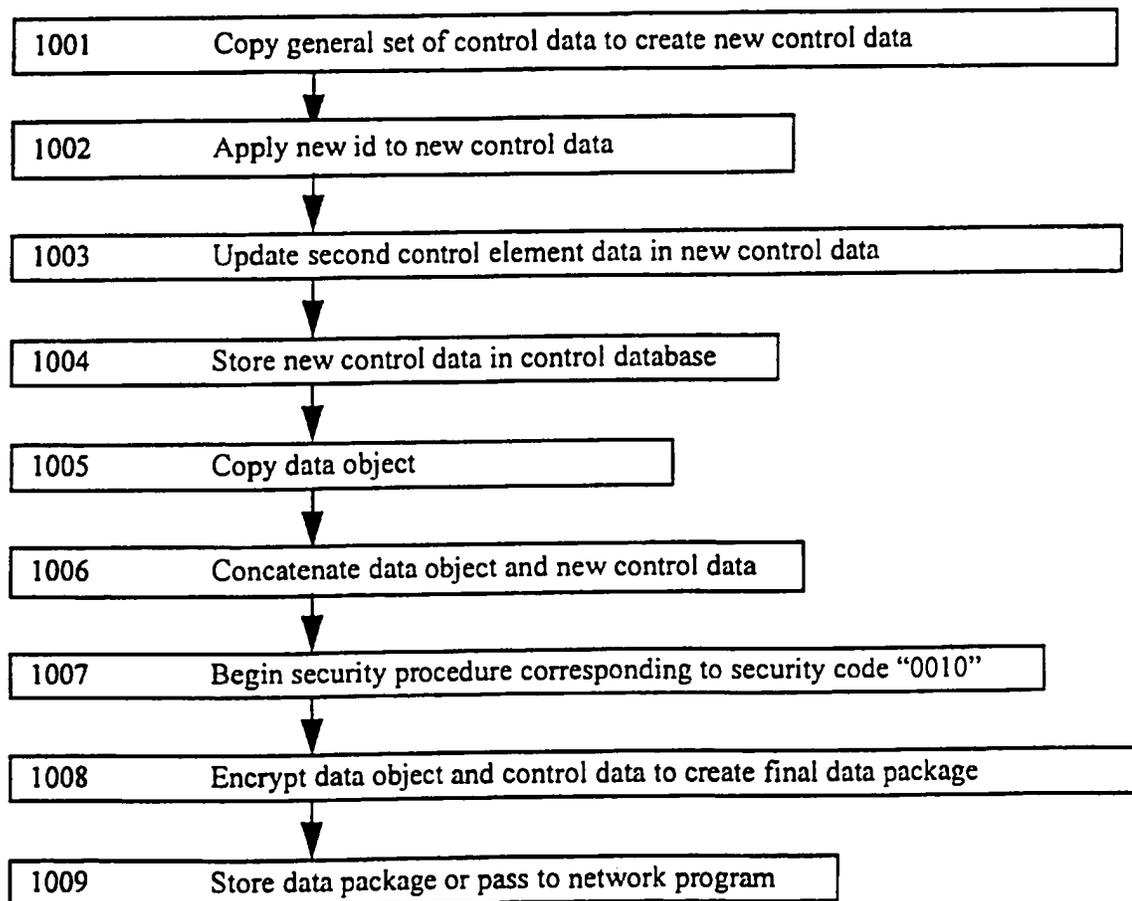
Fig 9



SUBSTITUTE SHEET

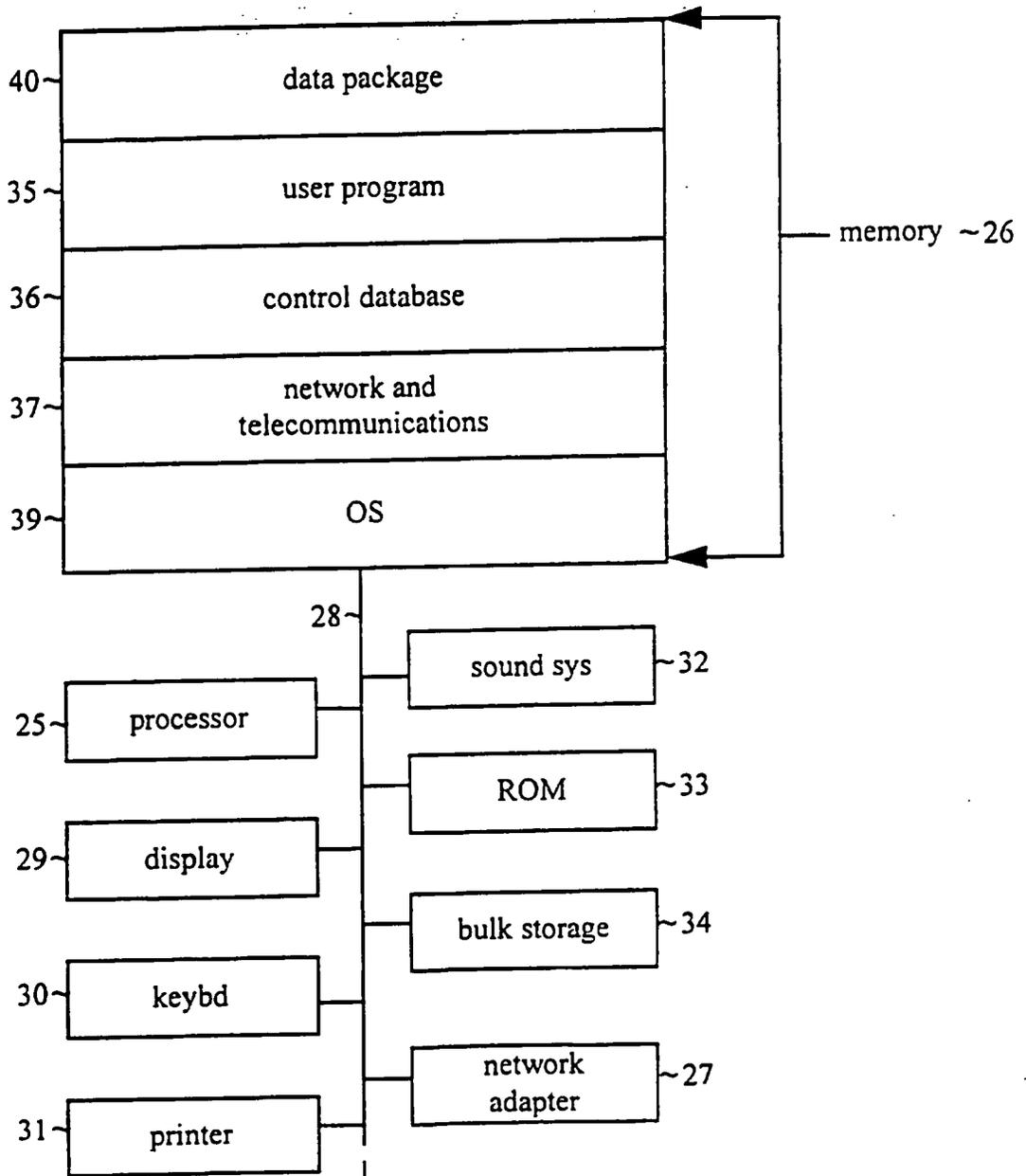
9/15

Fig 10

**SUBSTITUTE SHEET**

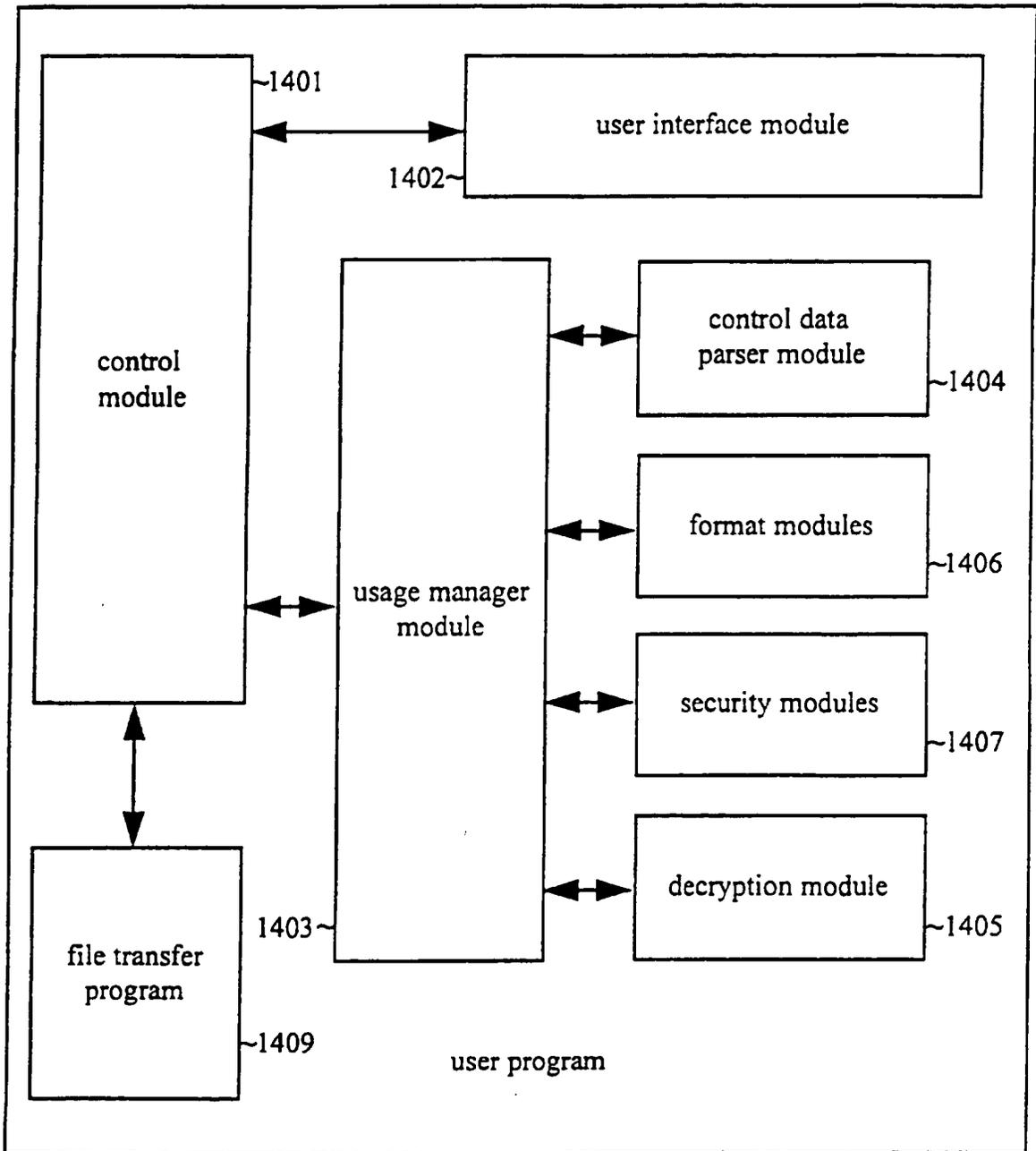
11/15

Fig 13



12/15

Fig 14

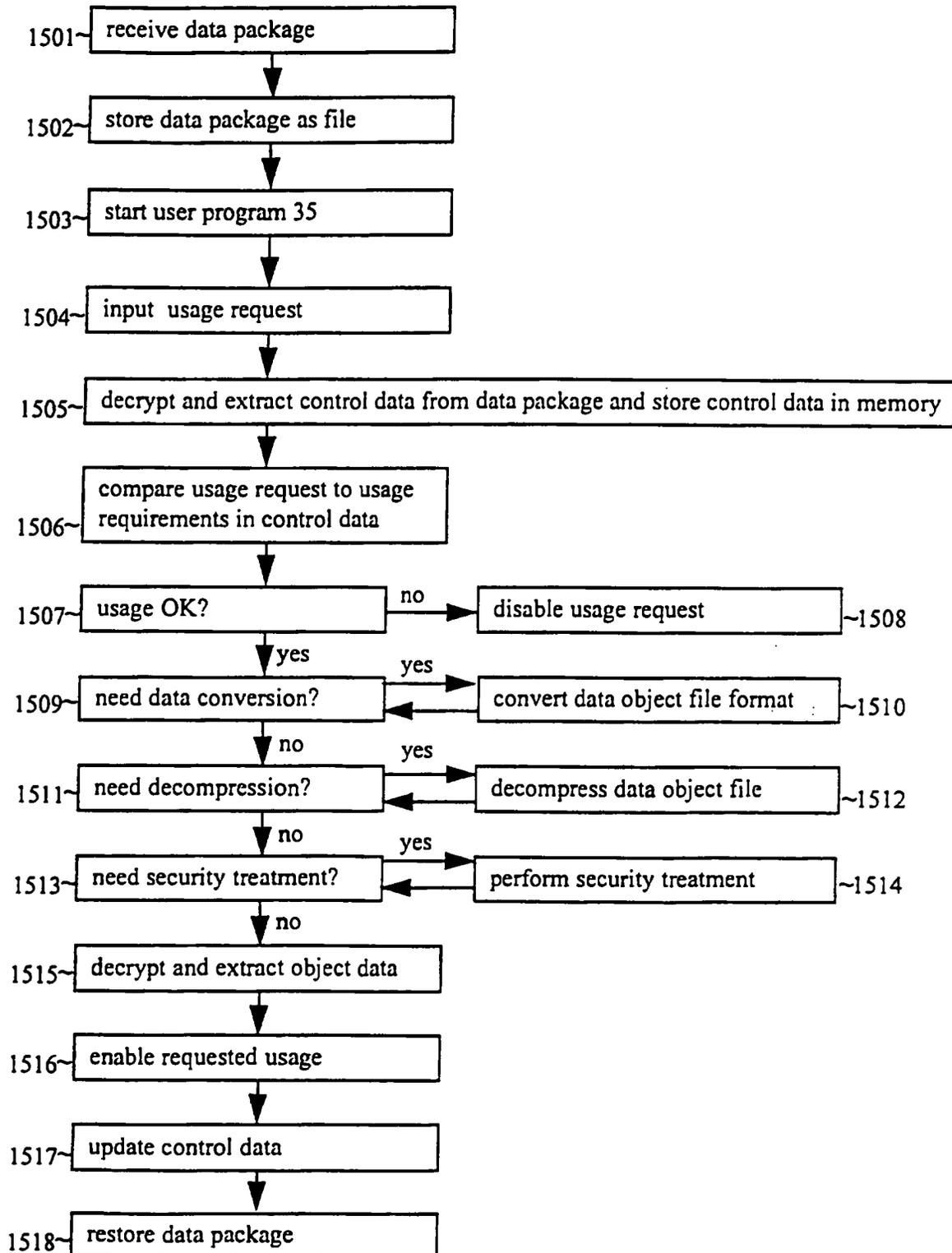


35

SUBSTITUTE SHEET

13/15

Fig 15



14/15

Fig 16

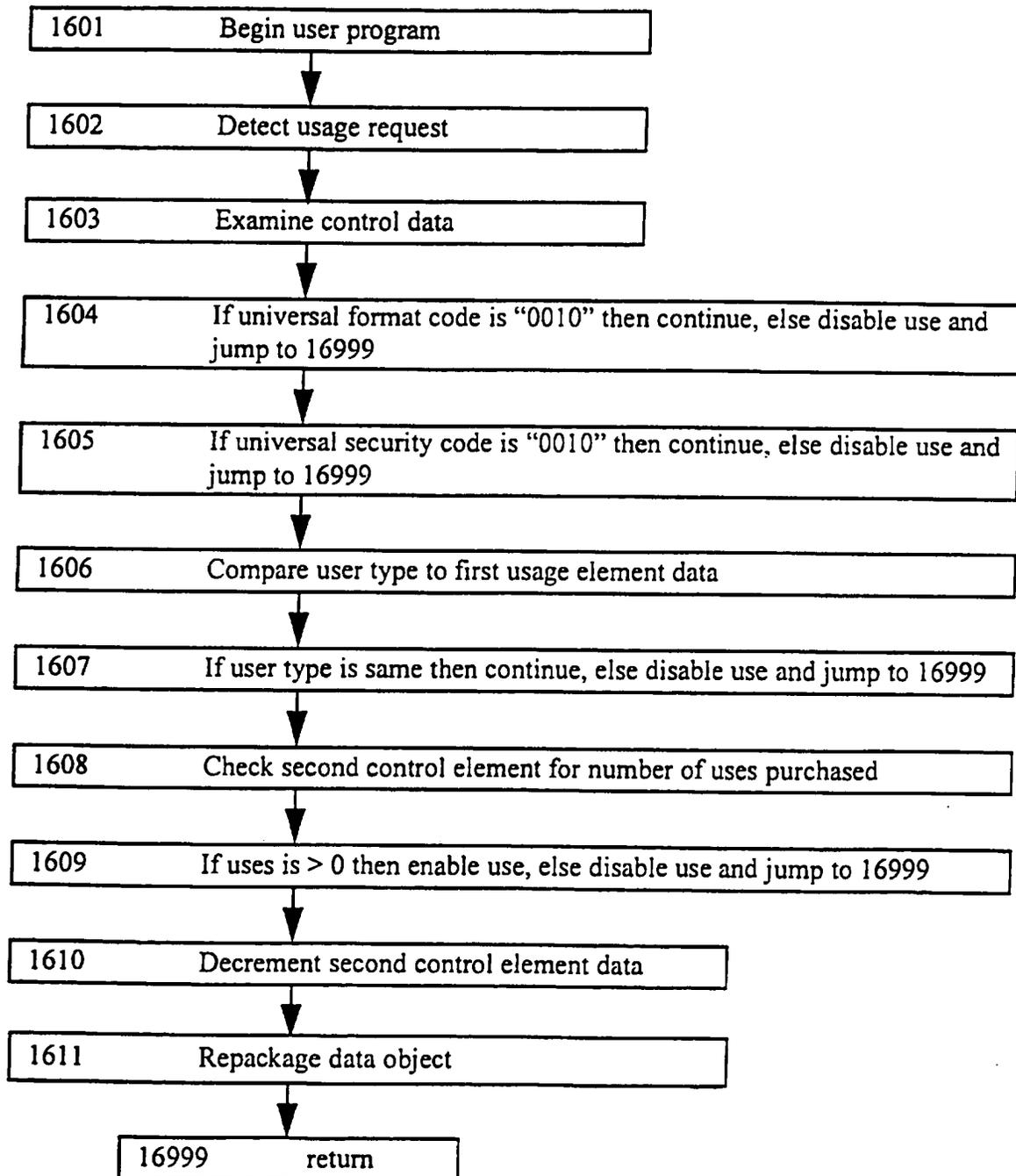
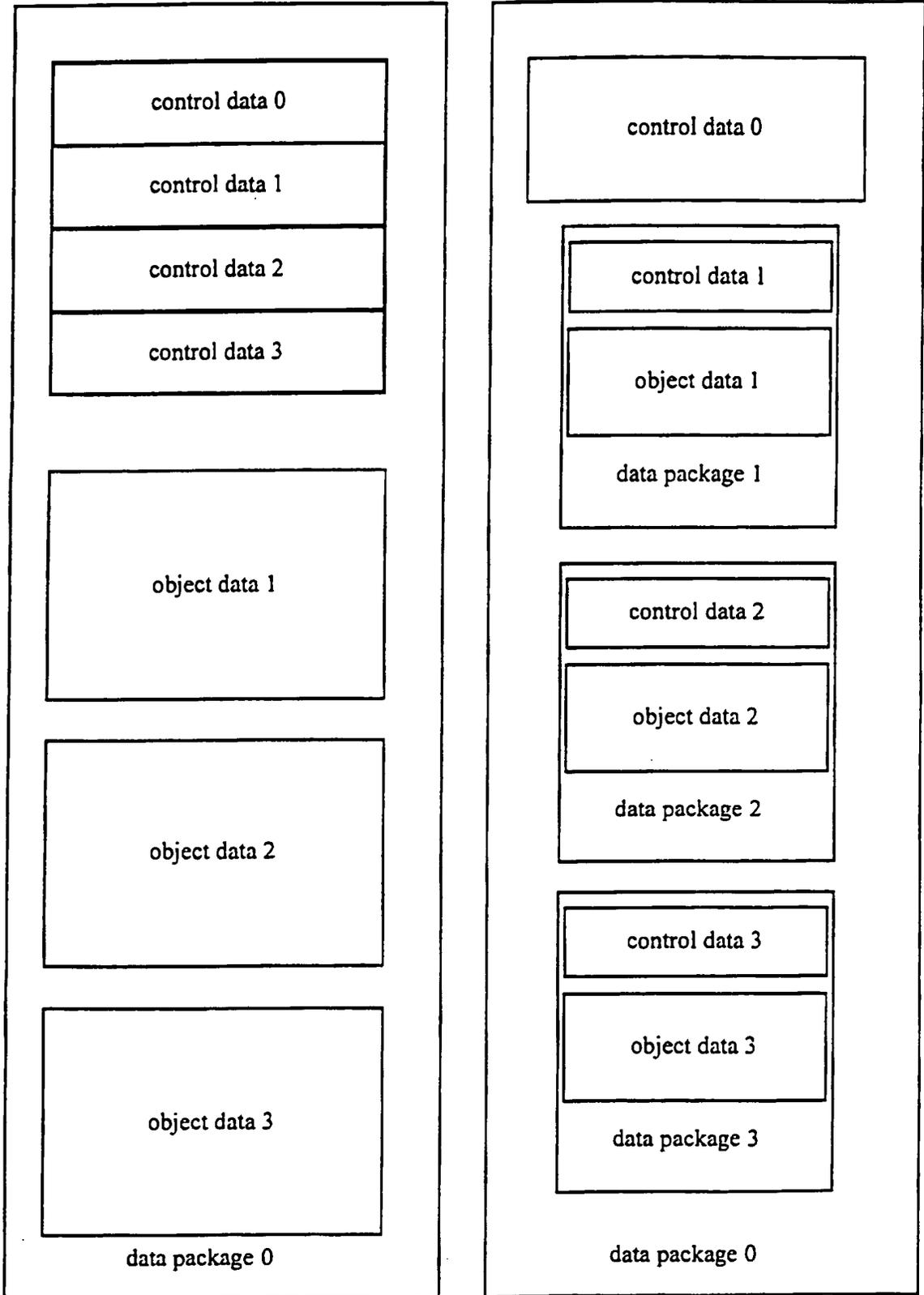
**SUBSTITUTE SHEET**

Fig 17



**CORRECTED
VERSION***

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁶ : G06F</p>	<p>A2</p>	<p>(11) International Publication Number: WO 96/27155 (43) International Publication Date: 6 September 1996 (06.09.96)</p>
<p>(21) International Application Number: PCT/US96/02303 (22) International Filing Date: 13 February 1996 (13.02.96) (30) Priority Data: 08/388,107 13 February 1995 (13.02.95) US (71) Applicant: ELECTRONIC PUBLISHING RESOURCES, INC. [US/US]; 5203 Battery Lane, Bethesda, MD 20814 (US). (72) Inventors: GINTER, Karl, L.; 10404 43rd Avenue, Beltsville, MD 20705 (US). SHEAR, Victor, H.; 5203 Battery Lane, Bethesda, MD 20814 (US). SPAHN, Francis, J.; 2410 Edwards Avenue, El Cerrito, CA 94530 (US). VAN WIE, David, M.; 1250 Lakeside Drive, Sunnyvale, CA 94086 (US). (74) Agent: FARIS, Robert, W.; Nixon & Vanderhye P.C., 1100 North Glebe Road, Arlington, VA 22201-4714 (US).</p>		<p>(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AZ, BY, KG, KZ, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).</p> <p>Published <i>Without international search report and to be republished upon receipt of that report.</i></p>
<p>(54) Title: SYSTEMS AND METHODS FOR SECURE TRANSACTION MANAGEMENT AND ELECTRONIC RIGHTS PROTECTION</p>		
<p>(57) Abstract</p> <p>The present invention provides systems and methods for electronic commerce including secure transaction management and electronic rights protection. Electronic appliances such as computers employed in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Secure subsystems used with such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions. Secure distributed and other operating system environments and architectures, employing, for example, secure semiconductor processing arrangements that may establish secure, protected environments at each node. These techniques may be used to support an end-to-end electronic information distribution capability that may be used, for example, utilizing the "electronic highway".</p>		

* (Referred to in PCT Gazette No. 52/1996, Section II)

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

**SYSTEMS AND METHODS FOR SECURE TRANSACTION
MANAGEMENT AND ELECTRONIC RIGHTS PROTECTION**

Field(s) of the Invention(s)

This invention generally relates to computer and/or
electronic security.

5

More particularly, this invention relates to systems and
techniques for secure transaction management. This invention
also relates to computer-based and other electronic appliance-
based technologies that help to ensure that information is
10 accessed and/or otherwise used only in authorized ways, and
maintains the integrity, availability, and/or confidentiality of
such information and processes related to such use.

10

The invention also relates to systems and methods for
15 protecting rights of various participants in electronic commerce
and other electronic or electronically-facilitated transactions.

15

The invention also relates to secure chains of handling and
control for both information content and information employed to
20 regulate the use of such content and consequences of such use. It
also relates to systems and techniques that manage, including
meter and/or limit and/or otherwise monitor use of electronically
stored and/or disseminated information. The invention

20

particularly relates to transactions, conduct and arrangements that make use of, including consequences of use of, such systems and/or techniques.

5 The invention also relates to distributed and other operating systems, environments and architectures. It also generally relates to secure architectures, including, for example, tamper-resistant hardware-based processors, that can be used to establish security at each node of a distributed system.

10

Background and Summary of the Invention(s)

Telecommunications, financial transactions, government processes, business operations, entertainment, and personal business productivity all now depend on electronic appliances. Millions of these electronic appliances have been electronically connected together. These interconnected electronic appliances comprise what is increasingly called the "information highway." Many businesses, academicians, and government leaders are concerned about how to protect the rights of citizens and organizations who use this information (also "electronic" or

15

20

25

"digital") highway.

Electronic Content

Today, virtually anything that can be represented by words, numbers, graphics, or system of commands and

instructions can be formatted into electronic digital information. Television, cable, satellite transmissions, and on-line services transmitted over telephone lines, compete to distribute digital information and entertainment to homes and businesses. The owners and marketers of this content include software developers, motion picture and recording companies, publishers of books, magazines, and newspapers, and information database providers. The popularization of on-line services has also enabled the individual personal computer user to participate as a content provider. It is estimated that the worldwide market for electronic information in 1992 was approximately \$40 billion and is expected to grow to \$200 billion by 1997, according to Microsoft Corporation. The present invention can materially enhance the revenue of content providers, lower the distribution costs and the costs for content, better support advertising and usage information gathering, and better satisfy the needs of electronic information users. These improvements can lead to a significant increase in the amount and variety of electronic information and the methods by which such information is distributed.

The inability of conventional products to be shaped to the needs of electronic information providers and users is sharply in contrast to the present invention. Despite the attention devoted by a cross-section of America's largest telecommunications, computer, entertainment and information provider companies to

some of the problems addressed by the present invention, only the present invention provides commercially secure, effective solutions for configurable, general purpose electronic commerce transaction/distribution control systems.

5

Controlling Electronic Content

The present invention provides a new kind of "virtual distribution environment" (called "VDE" in this document) that secures, administers, and audits electronic information use. VDE also features fundamentally important capabilities for managing content that travels "across" the "information highway." These capabilities comprise a rights protection solution that serves all electronic community members. These members include content creators and distributors, financial service providers, end-users, and others. VDE is the first general purpose, configurable, transaction control/rights protection solution for users of computers, other electronic appliances, networks, and the information highway.

15
20

A fundamental problem for electronic content providers is extending their ability to control the use of proprietary information. Content providers often need to limit use to authorized activities and amounts. Participants in a business model involving, for example, provision of movies and advertising on optical discs may include actors, directors, script and other

25

writers, musicians, studios, publishers, distributors, retailers, advertisers, credit card services, and content end-users. These participants need the ability to embody their range of agreements and requirements, including use limitations, into an "extended" agreement comprising an overall electronic business model. This extended agreement is represented by electronic content control information that can automatically enforce agreed upon rights and obligations. Under VDE, such an extended agreement may comprise an electronic contract involving all business model participants. Such an agreement may alternatively, or in addition, be made up of electronic agreements between subsets of the business model participants. Through the use of VDE, electronic commerce can function in the same way as traditional commerce—that is commercial relationships regarding products and services can be shaped through the negotiation of one or more agreements between a variety of parties.

Commercial content providers are concerned with ensuring proper compensation for the use of their electronic information. Electronic digital information, for example a CD recording, can today be copied relatively easily and inexpensively. Similarly, unauthorized copying and use of software programs deprives rightful owners of billions of dollars in annual revenue according to the International Intellectual Property Alliance. Content providers and distributors have devised a number of limited

function rights protection mechanisms to protect their rights. Authorization passwords and protocols, license servers, “lock/unlock” distribution methods, and non-electronic contractual limitations imposed on users of shrink-wrapped software are a few of the more prevalent content protection schemes. In a commercial context, these efforts are inefficient and limited solutions.

Providers of “electronic currency” have also created protections for their type of content. These systems are not sufficiently adaptable, efficient, nor flexible enough to support the generalized use of electronic currency. Furthermore, they do not provide sophisticated auditing and control configuration capabilities. This means that current electronic currency tools lack the sophistication needed for many real-world financial business models. VDE provides means for anonymous currency and for “conditionally” anonymous currency, wherein currency related activities remain anonymous except under special circumstances.

20

VDE Control Capabilities

VDE allows the owners and distributors of electronic digital information to reliably bill for, and securely control, audit, and budget the use of, electronic information. It can reliably

25

detect and monitor the use of commercial information products.

VDE uses a wide variety of different electronic information delivery means: including, for example, digital networks, digital broadcast, and physical storage media such as optical and magnetic disks. VDE can be used by major network providers, hardware manufacturers, owners of electronic information, providers of such information, and clearinghouses that gather usage information regarding, and bill for the use of, electronic information.

10

VDE provides comprehensive and configurable transaction management, metering and monitoring technology. It can change how electronic information products are protected, marketed, packaged, and distributed. When used, VDE should result in higher revenues for information providers and greater user satisfaction and value. Use of VDE will normally result in lower usage costs, decreased transaction costs, more efficient access to electronic information, re-usability of rights protection and other transaction management implementations, greatly improved flexibility in the use of secured information, and greater standardization of tools and processes for electronic transaction management. VDE can be used to create an adaptable environment that fulfills the needs of electronic information owners, distributors, and users; financial clearinghouses; and usage information analyzers and resellers.

15

20

25

Rights and Control Information

In general, the present invention can be used to protect the rights of parties who have:

- 5 (a) proprietary or confidentiality interests in electronic information. It can, for example, help ensure that information is used only in authorized ways;
- 10 (b) financial interests resulting from the use of electronically distributed information. It can help ensure that content providers will be paid for use of distributed information; and
- 15 (c) interests in electronic credit and electronic currency storage, communication, and/or use including electronic cash, banking, and purchasing.

Protecting the rights of electronic community members involves a broad range of technologies. VDE combines these technologies in a way that creates a "distributed" electronic rights protection "environment." This environment secures and protects transactions and other processes important for rights protection. VDE, for example, provides the ability to prevent, or impede, interference with and/or observation of, important rights related transactions and processes. VDE, in its preferred

20

25

embodiment, uses special purpose tamper resistant Secure Processing Units (SPUs) to help provide a high level of security for VDE processes and information storage and communication.

5 The rights protection problems solved by the present invention are electronic versions of basic societal issues. These issues include protecting property rights, protecting privacy rights, properly compensating people and organizations for their work and risk, protecting money and credit, and generally
10 protecting the security of information. VDE employs a system that uses a common set of processes to manage rights issues in an efficient, trusted, and cost-effective way.

 VDE can be used to protect the rights of parties who create
15 electronic content such as, for example: records, games, movies, newspapers, electronic books and reference materials, personal electronic mail, and confidential records and communications. The invention can also be used to protect the rights of parties who provide electronic products, such as publishers and
20 distributors; the rights of parties who provide electronic credit and currency to pay for use of products, for example, credit clearinghouses and banks; the rights to privacy of parties who use electronic content (such as consumers, business people, governments); and the privacy rights of parties described by
25 electronic information, such as privacy rights related to

information contained in a medical record, tax record, or
personnel record.

5 In general, the present invention can protect the rights of
parties who have:

(a) commercial interests in electronically distributed
information -- the present invention can help ensure,
for example, that parties, will be paid for use of
distributed information in a manner consistent with
10 their agreement;

(b) proprietary and/or confidentiality interests in
electronic information -- the present invention can,
for example, help ensure that data is used only in
15 authorized ways;

(c) interests in electronic credit and electronic currency
storage, communication, and/or use -- this can
include electronic cash, banking, and purchasing;
20 and

(d) interests in electronic information derived, at least
in part, from use of other electronic information.

25 **VDE Functional Properties**

VDE is a cost-effective and efficient rights protection solution that provides a unified, consistent system for securing and managing transaction processing. VDE can:

- 5 (a) audit and analyze the use of content,
- (b) ensure that content is used only in authorized ways,
 and
- 10 (c) allow information regarding content usage to be used
 only in ways approved by content users.

In addition, VDE:

- 15 (a) is very configurable, modifiable, and re-usable;
- (b) supports a wide range of useful capabilities that may
 be combined in different ways to accommodate most
 potential applications;
- 20 (c) operates on a wide variety of electronic appliances
 ranging from hand-held inexpensive devices to large
 mainframe computers;

- (d) is able to ensure the various rights of a number of different parties, and a number of different rights protection schemes, simultaneously;
- 5 (e) is able to preserve the rights of parties through a series of transactions that may occur at different times and different locations;
- 10 (f) is able to flexibly accommodate different ways of securely delivering information and reporting usage; and
- 15 (g) provides for electronic analogues to "real" money and credit, including anonymous electronic cash, to pay for products and services and to support personal (including home) banking and other financial activities.

VDE economically and efficiently fulfills the rights protection needs of electronic community members. Users of VDE will not require additional rights protection systems for different information highway products and rights problems—nor will they be required to install and learn a new system for each new information highway application.

25

VDE provides a unified solution that allows all content creators, providers, and users to employ the same electronic rights protection solution. Under authorized circumstances, the participants can freely exchange content and associated content control sets. This means that a user of VDE may, if allowed, use the same electronic system to work with different kinds of content having different sets of content control information. The content and control information supplied by one group can be used by people who normally use content and control information supplied by a different group. VDE can allow content to be exchanged "universally" and users of an implementation of the present invention can interact electronically without fear of incompatibilities in content control, violation of rights, or the need to get, install, or learn a new content control system.

The VDE securely administers transactions that specify protection of rights. It can protect electronic rights including, for example:

- (a) the property rights of authors of electronic content,
- (b) the commercial rights of distributors of content,
- (c) the rights of any parties who facilitated the distribution of content,

- (d) the privacy rights of users of content,
- (e) the privacy rights of parties portrayed by stored and/or distributed content, and
- 5 (f) any other rights regarding enforcement of electronic agreements.

10 VDE can enable a very broad variety of electronically enforced commercial and societal agreements. These agreements can include electronically implemented contracts, licenses, laws, regulations, and tax collection.

Contrast With Traditional Solutions

15 Traditional content control mechanisms often require users to purchase more electronic information than the user needs or desires. For example, infrequent users of shrink-wrapped software are required to purchase a program at the same price as frequent users, even though they may receive much less value
20 from their less frequent use. Traditional systems do not scale cost according to the extent or character of usage and traditional systems can not attract potential customers who find that a fixed price is too high. Systems using traditional mechanisms are also not normally particularly secure. For example, shrink-wrapping

does not prevent the constant illegal pirating of software once removed from either its physical or electronic package.

5 Traditional electronic information rights protection systems are often inflexible and inefficient and may cause a content provider to choose costly distribution channels that increase a product's price. In general these mechanisms restrict product pricing, configuration, and marketing flexibility. These compromises are the result of techniques for controlling
10 information which cannot accommodate both different content models and content models which reflect the many, varied requirements, such as content delivery strategies, of the model participants. This can limit a provider's ability to deliver sufficient overall value to justify a given product's cost in the eyes
15 of many potential users. VDE allows content providers and distributors to create applications and distribution networks that reflect content providers' and users' preferred business models. It offers users a uniquely cost effective and feature rich system that supports the ways providers want to distribute information
20 and the ways users want to use such information. VDE supports content control models that ensure rights and allow content delivery strategies to be shaped for maximum commercial results.

Chain of Handling and Control

VDE can protect a collection of rights belonging to various parties having in rights in, or to, electronic information. This information may be at one location or dispersed across (and/or moving between) multiple locations. The information may pass through a "chain" of distributors and a "chain" of users. Usage information may also be reported through one or more "chains" of parties. In general, VDE enables parties that (a) have rights in electronic information, and/or (b) act as direct or indirect agents for parties who have rights in electronic information, to ensure that the moving, accessing, modifying, or otherwise using of information can be securely controlled by rules regarding how, when, where, and by whom such activities can be performed.

VDE Applications and Software

VDE is a secure system for regulating electronic conduct and commerce. Regulation is ensured by control information put in place by one or more parties. These parties may include content providers, electronic hardware manufacturers, financial service providers, or electronic "infrastructure" companies such as cable or telecommunications companies. The control information implements "Rights Applications." Rights applications "run on" the "base software" of the preferred embodiment. This base software serves as a secure, flexible, general purpose foundation that can accommodate many

different rights applications, that is, many different business models and their respective participant requirements.

5 A rights application under VDE is made up of special purpose pieces, each of which can correspond to one or more basic electronic processes needed for a rights protection environment. These processes can be combined together like building blocks to create electronic agreements that can protect the rights, and may enforce fulfillment of the obligations, of electronic information users and providers. One or more providers of electronic information can easily combine selected building blocks to create a rights application that is unique to a specific content distribution model. A group of these pieces can represent the capabilities needed to fulfill the agreement(s) between users and providers. These pieces accommodate many requirements of electronic commerce including:

10

15

! the distribution of permissions to use electronic information;

20

! the persistence of the control information and sets of control information managing these permissions;

25

! configurable control set information that can be selected by users for use with such information;

! data security and usage auditing of electronic
information; and

5

! a secure system for currency, compensation and
debit management.

10

For electronic commerce, a rights application, under the preferred embodiment of the present invention, can provide electronic enforcement of the business agreements between all participants. Since different groups of components can be put together for different applications, the present invention can provide electronic control information for a wide variety of different products and markets. This means the present invention can provide a "unified," efficient, secure, and cost-effective system for electronic commerce and data security. This allows VDE to serve as a single standard for electronic rights protection, data security, and electronic currency and banking.

15

20

In a VDE, the separation between a rights application and its foundation permits the efficient selection of sets of control information that are appropriate for each of many different types of applications and uses. These control sets can reflect both rights of electronic community members, as well as obligations (such as providing a history of one's use of a product or paying

25

taxes on one's electronic purchases). VDE flexibility allows its users to electronically implement and enforce common social and commercial ethics and practices. By providing a unified control system, the present invention supports a vast range of possible transaction related interests and concerns of individuals, communities, businesses, and governments. Due to its open design, VDE allows (normally under securely controlled circumstances) applications using technology independently created by users to be "added" to the system and used in conjunction with the foundation of the invention. In sum, VDE provides a system that can fairly reflect and enforce agreements among parties. It is a broad ranging and systematic solution that answers the pressing need for a secure, cost-effective, and fair electronic environment.

15

VDE Implementation

The preferred embodiment of the present invention includes various tools that enable system designers to directly insert VDE capabilities into their products. These tools include an Application Programmer's Interface ("API") and a Rights Permissioning and Management Language ("RPML"). The RPML provides comprehensive and detailed control over the use of the invention's features. VDE also includes certain user interface subsystems for satisfying the needs of content providers, distributors, and users.

25

Information distributed using VDE may take many forms. It may, for example, be "distributed" for use on an individual's own computer, that is the present invention can be used to provide security for locally stored data. Alternatively, VDE may
5 be used with information that is dispersed by authors and/or publishers to one or more recipients. This information may take many forms including: movies, audio recordings, games, electronic catalog shopping, multimedia, training materials, E-mail and personal documents, object oriented libraries,
10 software programming resources, and reference/record keeping information resources (such as business, medical, legal, scientific, governmental, and consumer databases).

Electronic rights protection provided by the present
15 invention will also provide an important foundation for trusted and efficient home and commercial banking, electronic credit processes, electronic purchasing, true or conditionally anonymous electronic cash, and EDI (Electronic Data Interchange). VDE provides important enhancements for improving data security in
20 organizations by providing "smart" transaction management features that can be far more effective than key and password based "go/no go" technology.

VDE normally employs an integration of cryptographic and
25 other security technologies (e.g. encryption, digital signatures,

etc.), with other technologies including: component, distributed, and event driven operating system technology, and related communications, object container, database, smart agent, smart card, and semiconductor design technologies.

5

I. Overview

A. VDE Solves Important Problems and Fills Critical Needs

The world is moving towards an integration of electronic information appliances. This interconnection of appliances provides a foundation for much greater electronic interaction and the evolution of electronic commerce. A variety of capabilities are required to implement an electronic commerce environment. VDE is the first system that provides many of these capabilities and therefore solves fundamental problems related to electronic dissemination of information.

10
15

Electronic Content

VDE allows electronic arrangements to be created involving two or more parties. These agreements can themselves comprise a collection of agreements between participants in a commercial value chain and/or a data security chain model for handling, auditing, reporting, and payment. It can provide efficient, reusable, modifiable, and consistent means for secure electronic content: distribution, usage control, usage payment,

20
25

usage auditing, and usage reporting. Content may, for example, include:

5 ! financial information such as electronic currency and credit;

 ! commercially distributed electronic information such as reference databases, movies, games, and advertising; and

10 ! electronic properties produced by persons and organizations, such as documents, e-mail, and proprietary database information.

15 VDE enables an electronic commerce marketplace that supports differing, competitive business partnerships, agreements, and evolving overall business models.

 The features of VDE allow it to function as the first trusted
20 electronic information control environment that can conform to, and support, the bulk of conventional electronic commerce and data security requirements. In particular, VDE enables the participants in a business value chain model to create an
25 electronic version of traditional business agreement terms and conditions and further enables these participants to shape and

evolve their electronic commerce models as they believe appropriate to their business requirements.

5 VDE offers an architecture that avoids reflecting specific distribution biases, administrative and control perspectives, and content types. Instead, VDE provides a broad-spectrum, fundamentally configurable and portable, electronic transaction control, distributing, usage, auditing, reporting, and payment operating environment. VDE is not limited to being an application or application specific toolset that covers only a limited subset of electronic interaction activities and participants. Rather, VDE supports systems by which such applications can be created, modified, and/or reused. As a result, the present invention answers pressing, unsolved needs by offering a system that supports a standardized control environment which facilitates interoperability of electronic appliances, interoperability of content containers, and efficient creation of electronic commerce applications and models through the use of a programmable, secure electronic transactions management foundation and reusable and extensible executable components. VDE can support a single electronic "world" within which most forms of electronic transaction activities can be managed.

10
15
20

To answer the developing needs of rights owners and content providers and to provide a system that can accommodate

25

the requirements and agreements of all parties that may be involved in electronic business models (creators, distributors, administrators, users, credit providers, etc.), VDE supplies an efficient, largely transparent, low cost and sufficiently secure system (supporting both hardware/ software and software only models). VDE provides the widely varying secure control and administration capabilities required for:

1. Different types of electronic content,
2. Differing electronic content delivery schemes,
3. Differing electronic content usage schemes,
4. Different content usage platforms, and
5. Differing content marketing and model strategies.

VDE may be combined with, or integrated into, many separate computers and/or other electronic appliances. These appliances typically include a secure subsystem that can enable control of content use such as displaying, encrypting, decrypting, printing, copying, saving, extracting, embedding, distributing, auditing usage, etc. The secure subsystem in the preferred embodiment comprises one or more "protected processing

environments", one or more secure databases, and secure
"component assemblies" and other items and processes that need
to be kept secured. VDE can, for example, securely control
electronic currency, payments, and/or credit management
5 (including electronic credit and/or currency receipt,
disbursement, encumbering, and/or allocation) using such a
"secure subsystem."

VDE provides a secure, distributed electronic transaction
10 management system for controlling the distribution and/or other
usage of electronically provided and/or stored information. VDE
controls auditing and reporting of electronic content and/or
appliance usage. Users of VDE may include content creators who
apply content usage, usage reporting, and/or usage payment
15 related control information to electronic content and/or
appliances for users such as end-user organizations, individuals,
and content and/or appliance distributors. VDE also securely
supports the payment of money owed (including money owed for
content and/or appliance usage) by one or more parties to one or
20 more other parties, in the form of electronic credit and/or
currency.

Electronic appliances under control of VDE represent VDE
'nodes' that securely process and control; distributed electronic
25 information and/or appliance usage, control information

formulation, and related transactions. VDE can securely manage
the integration of control information provided by two or more
parties. As a result, VDE can construct an electronic agreement
between VDE participants that represent a "negotiation"
5 between, the control requirements of, two or more parties and
enacts terms and conditions of a resulting agreement. VDE
ensures the rights of each party to an electronic agreement
regarding a wide range of electronic activities related to
electronic information and/or appliance usage.

10

Through use of VDE's control system, traditional content
providers and users can create electronic relationships that
reflect traditional, non-electronic relationships. They can shape
and modify commercial relationships to accommodate the
15 evolving needs of, and agreements among, themselves. VDE does
not require electronic content providers and users to modify their
business practices and personal preferences to conform to a
metering and control application program that supports limited,
largely fixed functionality. Furthermore, VDE permits
20 participants to develop business models not feasible with non-
electronic commerce, for example, involving detailed reporting of
content usage information, large numbers of distinct transactions
at hitherto infeasibly low price points, "pass-along" control
information that is enforced without involvement or advance
25 knowledge of the participants, etc.

The present invention allows content providers and users
to formulate their transaction environment to accommodate:

- 5 (1) desired content models, content control models, and
content usage information pathways,
- (2) a complete range of electronic media and distribution
means,
- 10 (3) a broad range of pricing, payment, and auditing
strategies,
- (4) very flexible privacy and/or reporting models,
- 15 (5) practical and effective security architectures, and
- (6) other administrative procedures that together with
steps (1) through (5) can enable most "real world"
electronic commerce and data security models,
20 including models unique to the electronic world.

VDE's transaction management capabilities can enforce:

- (1) privacy rights of users related to information regarding their usage of electronic information and/or appliances,
- 5 (2) societal policy such as laws that protect rights of content users or require the collection of taxes derived from electronic transaction revenue, and
- 10 (3) the proprietary and/or other rights of parties related to ownership of, distribution of, and/or other commercial rights related to, electronic information.

VDE can support "real" commerce in an electronic form, that is the progressive creation of commercial relationships that form, over time, a network of interrelated agreements representing a value chain business model. This is achieved in part by enabling content control information to develop through the interaction of (negotiation between) securely created and independently submitted sets of content and/or appliance control information. Different sets of content and/or appliance control information can be submitted by different parties in an electronic business value chain enabled by the present invention. These parties create control information sets through the use of their respective VDE installations. Independently, securely deliverable, component based control information allows efficient

15

20

25

interaction among control information sets supplied by different parties.

5 VDE permits multiple, separate electronic arrangements to be formed between subsets of parties in a VDE supported electronic value chain model. These multiple agreements together comprise a VDE value chain "extended" agreement. VDE allows such constituent electronic agreements, and therefore overall VDE extended agreements, to evolve and
10 reshape over time as additional VDE participants become involved in VDE content and/or appliance control information handling. VDE electronic agreements may also be extended as new control information is submitted by existing participants. With VDE, electronic commerce participants are free to structure
15 and restructure their electronic commerce business activities and relationships. As a result, the present invention allows a competitive electronic commerce marketplace to develop since the use of VDE enables different, widely varying business models using the same or shared content.

20 A significant facet of the present invention's ability to broadly support electronic commerce is its ability to securely manage independently delivered VDE component objects containing control information (normally in the form of VDE
25 objects containing one or more methods, data, or load module

VDE components). This independently delivered control information can be integrated with senior and other pre-existing content control information to securely form derived control information using the negotiation mechanisms of the present invention. All requirements specified by this derived control information must be satisfied before VDE controlled content can be accessed or otherwise used. This means that, for example, all load modules and any mediating data which are listed by the derived control information as required must be available and securely perform their required function. In combination with other aspects of the present invention, securely, independently delivered control components allow electronic commerce participants to freely stipulate their business requirements and trade offs. As a result, much as with traditional, non-electronic commerce, the present invention allows electronic commerce (through a progressive stipulation of various control requirements by VDE participants) to evolve into forms of business that are the most efficient, competitive and useful.

VDE provides capabilities that rationalize the support of electronic commerce and electronic transaction management. This rationalization stems from the reusability of control structures and user interfaces for a wide variety of transaction management related activities. As a result, content usage control, data security, information auditing, and electronic

financial activities, can be supported with tools that are reusable, convenient, consistent, and familiar. In addition, a rational approach—a transaction/distribution control standard—allows all participants in VDE the same foundation set of hardware control and security, authoring, administration, and management tools to support widely varying types of information, business market model, and/or personal objectives.

Employing VDE as a general purpose electronic transaction/distribution control system allows users to maintain a single transaction management control arrangement on each of their computers, networks, communication nodes, and/or other electronic appliances. Such a general purpose system can serve the needs of many electronic transaction management applications without requiring distinct, different installations for different purposes. As a result, users of VDE can avoid the confusion and expense and other inefficiencies of different, limited purpose transaction control applications for each different content and/or business model. For example, VDE allows content creators to use the same VDE foundation control arrangement for both content authoring and for licensing content from other content creators for inclusion into their products or for other use. Clearinghouses, distributors, content creators, and other VDE users can all interact, both with the applications running on their VDE installations, and with each other, in an entirely consistent

manner, using and reusing (largely transparently) the same distributed tools, mechanisms, and consistent user interfaces, regardless of the type of VDE activity.

5 VDE prevents many forms of unauthorized use of electronic information, by controlling and auditing (and other administration of use) electronically stored and/or disseminated information. This includes, for example, commercially distributed content, electronic currency, electronic credit,
10 business transactions (such as EDI), confidential communications, and the like. VDE can further be used to enable commercially provided electronic content to be made available to users in user defined portions, rather than constraining the user to use portions of content that were "predetermined" by a content
15 creator and/or other provider for billing purposes.

VDE, for example, can employ:

- 20 (1) Secure metering means for budgeting and/or auditing electronic content and/or appliance usage;
- (2) Secure flexible means for enabling compensation and/or billing rates for content and/or appliance usage, including electronic credit and/or currency
25 mechanisms for payment means;

- 5
- (3) Secure distributed database means for storing control and usage related information (and employing validated compartmentalization and tagging schemes);
- 10
- (4) Secure electronic appliance control means;
- 15
- (5) A distributed, secure, "virtual black box" comprised of nodes located at every user (including VDE content container creators, other content providers, client users, and recipients of secure VDE content usage information) site. The nodes of said virtual black box normally include a secure subsystem having at least one secure hardware element (a semiconductor element or other hardware module for securely executing VDE control processes), said secure subsystems being distributed at nodes along a pathway of information storage, distribution, payment, usage, and/or auditing. In some
- 20
- embodiments, the functions of said hardware element, for certain or all nodes, may be performed by software, for example, in host processing environments of electronic appliances;
- 25
- (6) Encryption and decryption means;

- 5 (7) Secure communications means employing authentication, digital signaturing, and encrypted transmissions. The secure subsystems at said user nodes utilize a protocol that establishes and authenticates each node's and/or participant's identity, and establishes one or more secure host-to-host encryption keys for communications between the secure subsystems; and
- 10 (8) Secure control means that can allow each VDE installation to perform VDE content authoring (placing content into VDE containers with associated control information), content distribution, and content usage; as well as clearinghouse and other
- 15 administrative and analysis activities employing content usage information.

VDE may be used to migrate most non-electronic, traditional information delivery models (including entertainment, reference materials, catalog shopping, etc.) into an adequately secure digital distribution and usage management and payment context. The distribution and financial pathways managed by a VDE arrangement may include:

25 ! content creator(s),

- 5
- ! distributor(s),
 - ! redistributor(s),
 - ! client administrator(s),
 - ! client user(s),
 - ! financial and/or other clearinghouse(s),
 - ! and/or government agencies.

These distribution and financial pathways may also include:

- 10
- ! advertisers,
 - ! market survey organizations, and/or
 - ! other parties interested in the user usage of
information securely delivered and/or stored using
VDE.

15

Normally, participants in a VDE arrangement will employ the same secure VDE foundation. Alternate embodiments support VDE arrangements employing differing VDE foundations. Such alternate embodiments may employ procedures to ensure certain
20 interoperability requirements are met.

Secure VDE hardware (also known as SPUs for Secure Processing Units), or VDE installations that use software to substitute for, or complement, said hardware (provided by Host
25 Processing Environments (HPEs)), operate in conjunction with