[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 AID_1, AID_2, ········, AID_N, Enabler of AID_1, Enabler of AID_2, ········, 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 PAT_1 PAT_2 ········ 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 PAT_1 (AID_{11}) (AID_{21} AID_{22}) ········ (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 PAT_1 PAT_2 AID Enabler of AID_1 Enabler of AID_2

  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 (AID_{11}) (AID_{21} AID_{22}) ········ (AID_{N1} AID_{N2} ········ AID_{NM}) Enabler of AID

  the link informations of (AID_{11}), (AID_{21} AID_{22}) 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 eight embodiment can be carried out as follows.

[0338] First, the SCS generates an arbitrary information 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 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 (MakePAT) 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 $PAT<AID_{Null} \mid AID_{member1}, AID_{member2}, \cdots\cdots, AID_{memberN}>$ is allowed, but $PAT<AID_{holder} \mid AID_{Null}, AID_{member1}, AID_{member2}, \cdots\cdots, AID_{memberN}>$ is not allowed, so that the holder of $PAT<AID_{holder} \mid AID_{member}>$ cannot produce $PAT<AID_{Null} \mid AID_{member}>$ from this $PAT<AID_{holder} \mid AID_{member}>$ 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.

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.

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.

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.

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.

12. The method of claim 10, wherein the transfer control flag of the personalized access ticket is set by a trusted third party.

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.

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:

p robabilistically 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 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 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.

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 persnalized 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 devicer 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:

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 persnalized 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.

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:

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.

FIG.1

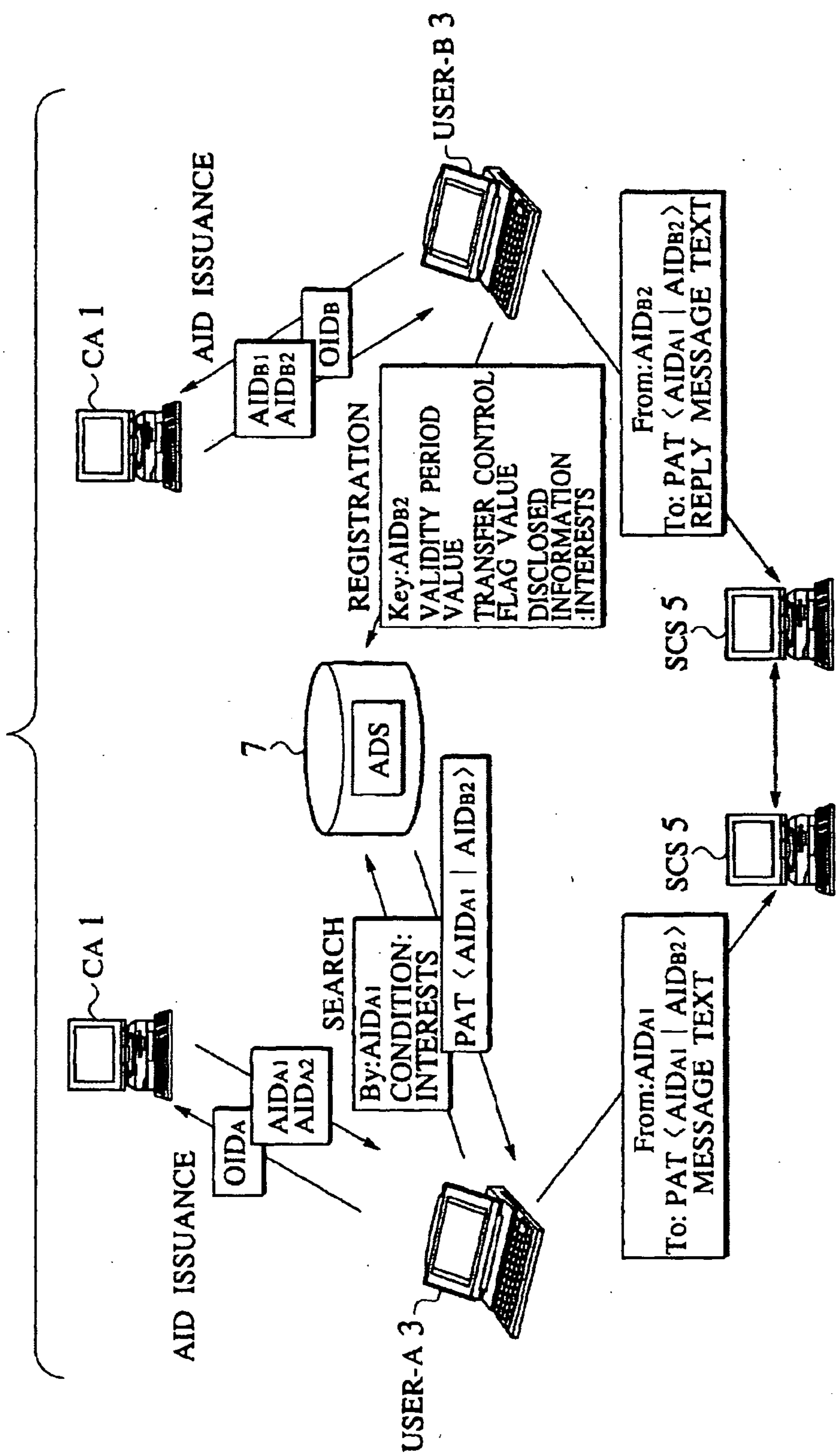

AID ISSUANCE
CA 1
OIDA
AIDA1
AIDA2
USER-A 3
SEARCH
By:AIDA1
CONDITION:
INTERESTS
PAT 〈AIDA1 | AIDB2〉
7
ADS
REGISTRATION
Key:AIDB2
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
DISCLOSED INFORMATION :INTERESTS
AID ISSUANCE
CA 1
AIDB1
AIDB2
OIDB
USER-B 3
From:AIDA1
To: PAT 〈AIDA1 | AIDB2〉
MESSAGE TEXT
From:AIDB2
To: PAT 〈AIDA1 | AIDB2〉
REPLY MESSAGE TEXT
SCS 5
SMTP
SCS 5
SMTP

FIG.2

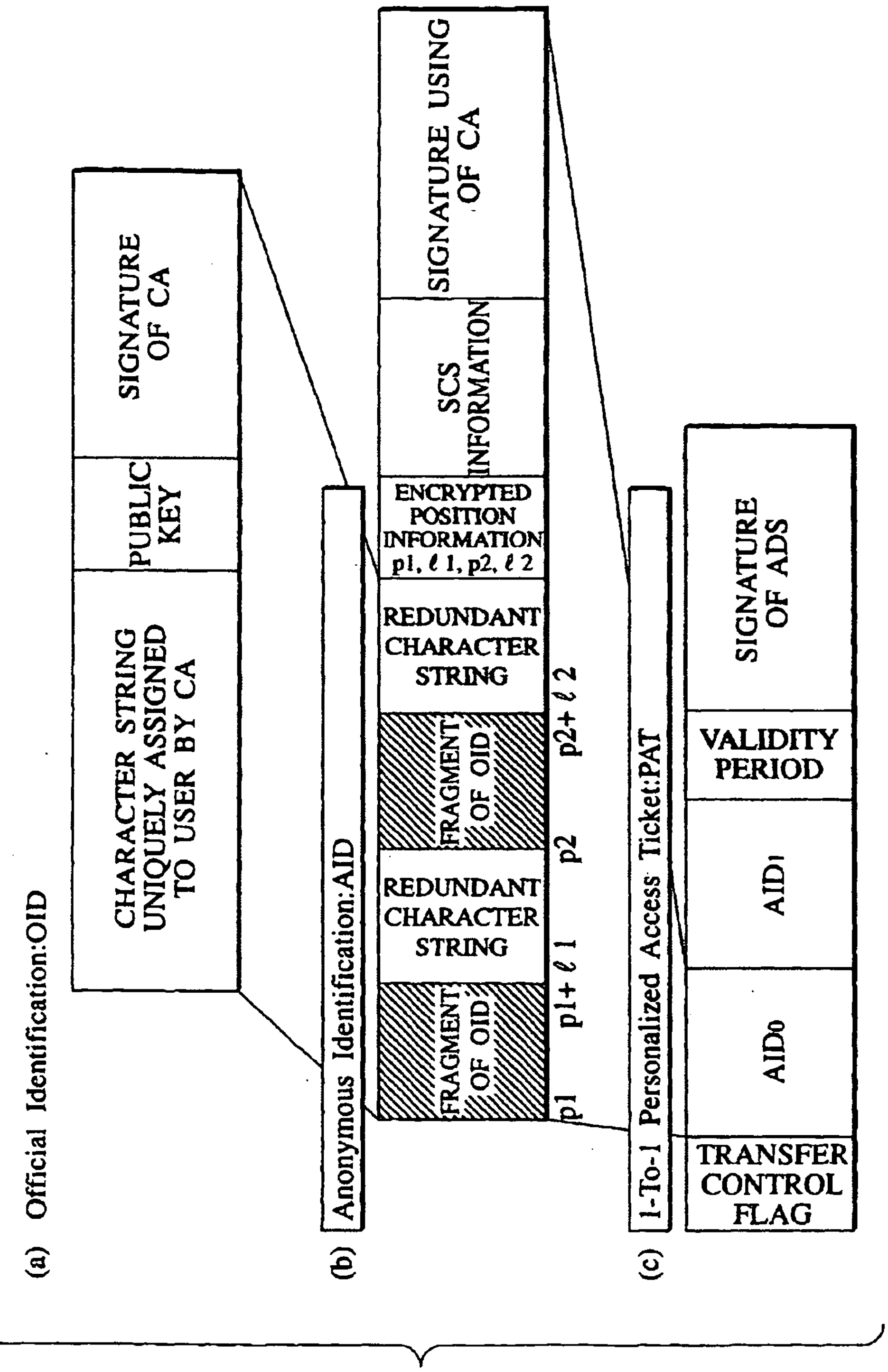

(a) Official Identification:OID
CHARACTER STRING UNIQUELY ASSIGNED TO USER BY CA
PUBLIC KEY
SIGNATURE OF CA
(b) Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
ENCRYPTED POSITION INFORMATION p1, ℓ1, p2, ℓ2
SCS INFORMATION
SIGNATURE USING OF CA
p1
p1+ ℓ1
p2
p2+ ℓ2
(c) 1-To-1 Personalized Access Ticket:PAT
TRANSFER CONTROL FLAG
AID0
AID1
VALIDITY PERIOD
SIGNATURE OF ADS

FIG.3

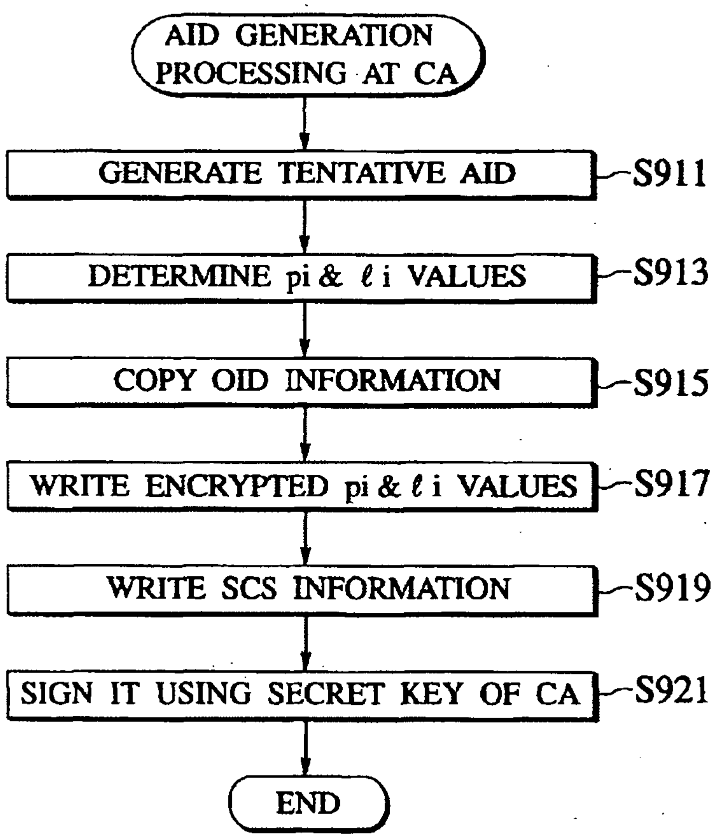

AID GENERATION PROCESSING AT CA
GENERATE TENTATIVE AID
S911
DETERMINE pi & ℓi VALUES
S913
COPY OID INFORMATION
S915
WRITE ENCRYPTED pi & ℓi VALUES
S917
WRITE SCS INFORMATION
S919
SIGN IT USING SECRET KEY OF CA
S921
END

FIG.4

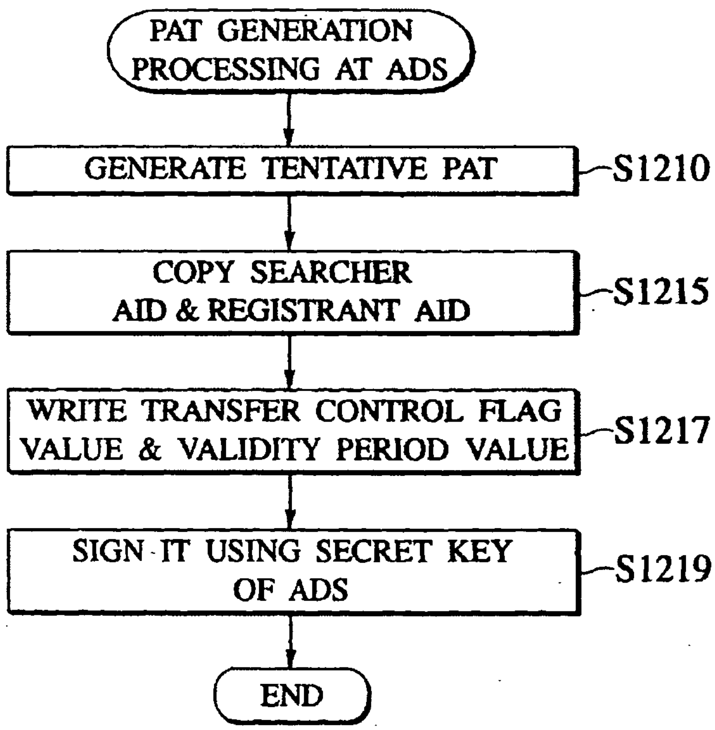

PAT GENERATION PROCESSING AT ADS
GENERATE TENTATIVE PAT
S1210
COPY SEARCHER AID & REGISTRANT AID
S1215
WRITE TRANSFER CONTROL FLAG VALUE & VALIDITY PERIOD VALUE
S1217
SIGN IT USING SECRET KEY OF ADS
S1219
END

FIG.5

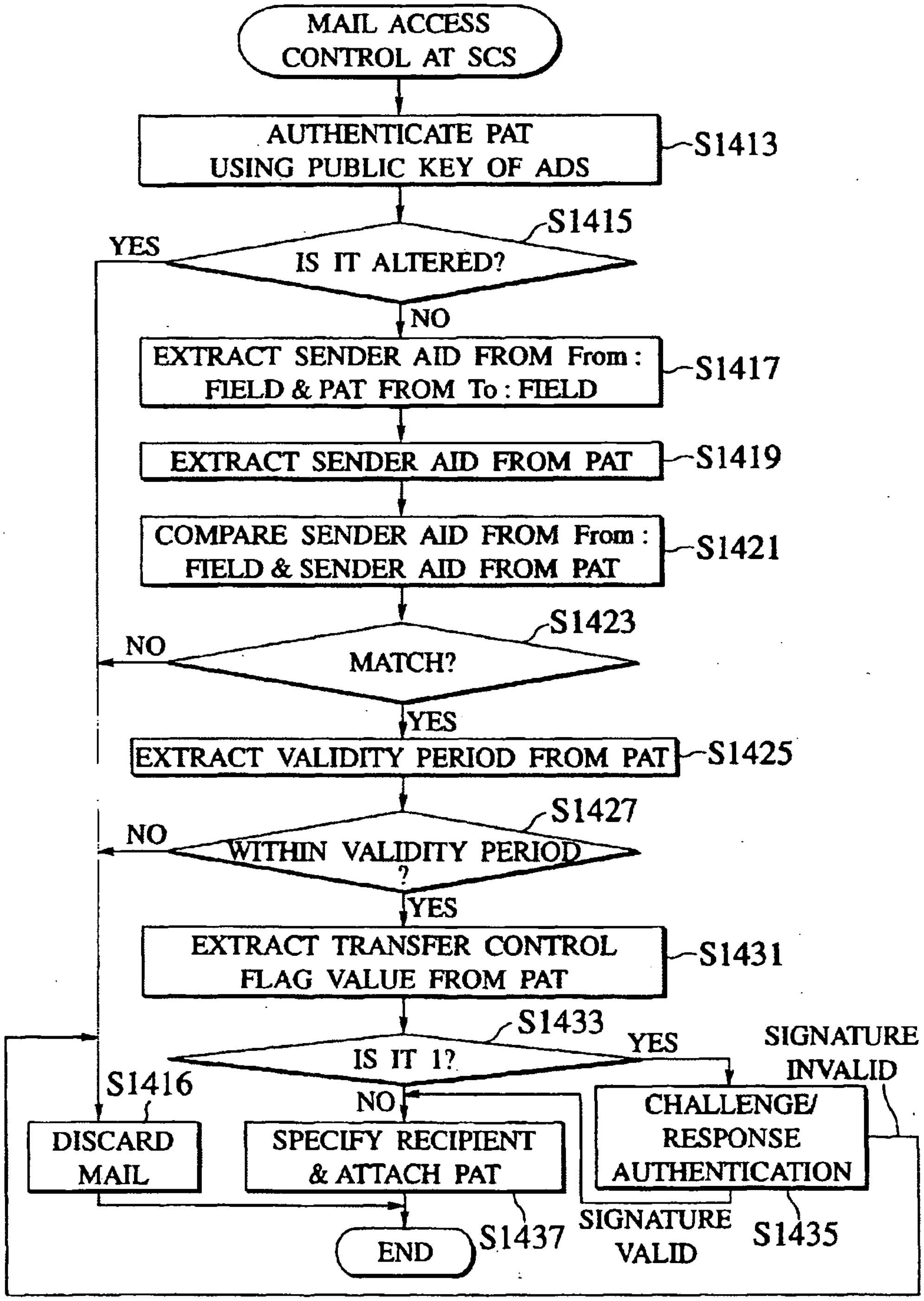

MAIL ACCESS CONTROL AT SCS
AUTHENTICATE PAT USING PUBLIC KEY OF ADS
S1413
S1415
IS IT ALTERED?
YES
NO
EXTRACT SENDER AID FROM From : FIELD & PAT FROM To : FIELD
S1417
EXTRACT SENDER AID FROM PAT
S1419
COMPARE SENDER AID FROM From : FIELD & SENDER AID FROM PAT
S1421
S1423
MATCH?
NO
YES
EXTRACT VALIDITY PERIOD FROM PAT
S1425
S1427
WITHIN VALIDITY PERIOD ?
NO
YES
EXTRACT TRANSFER CONTROL FLAG VALUE FROM PAT
S1431
S1433
IS IT 1?
YES
NO
SIGNATURE INVALID
S1416
DISCARD MAIL
SPECIFY RECIPIENT & ATTACH PAT
CHALLENGE/ RESPONSE AUTHENTICATION
END
S1437
SIGNATURE VALID
S1435

FIG.6

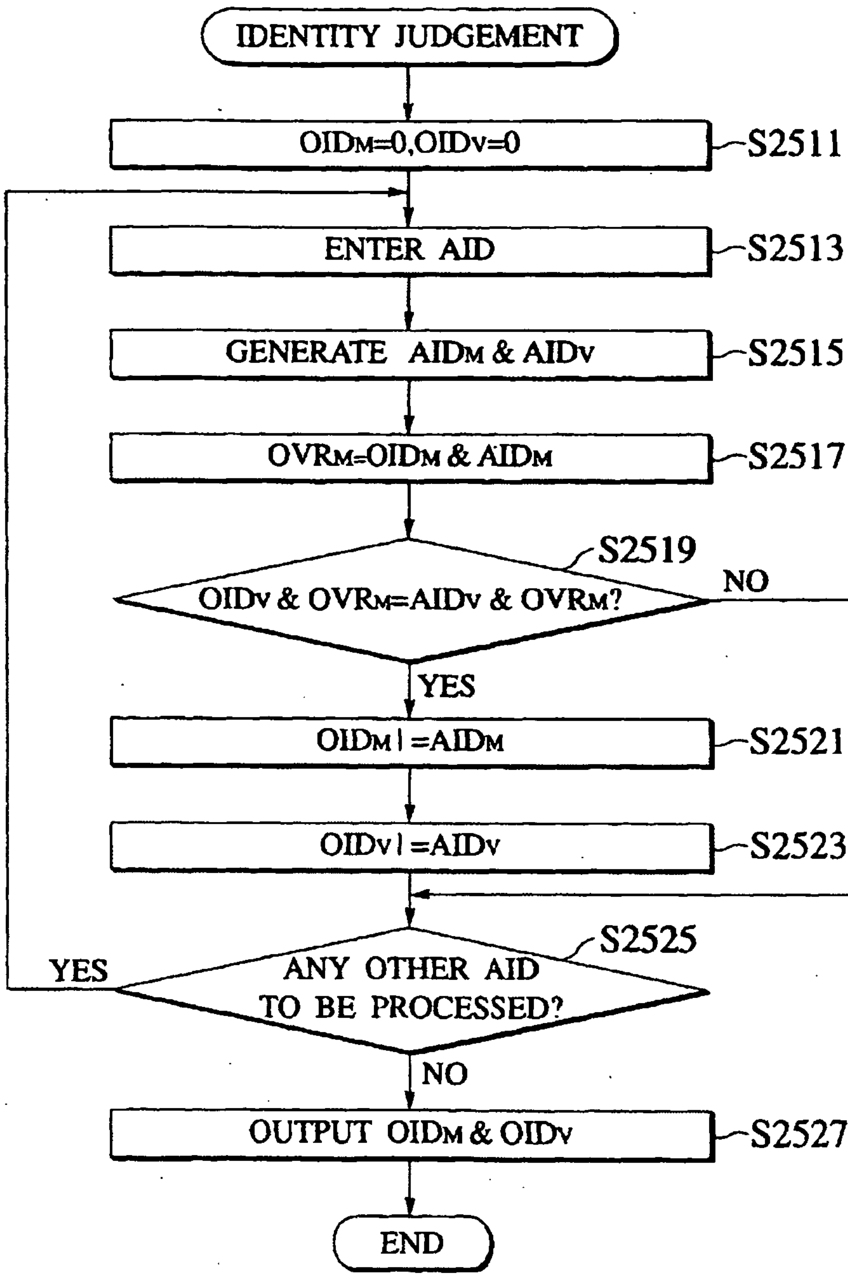
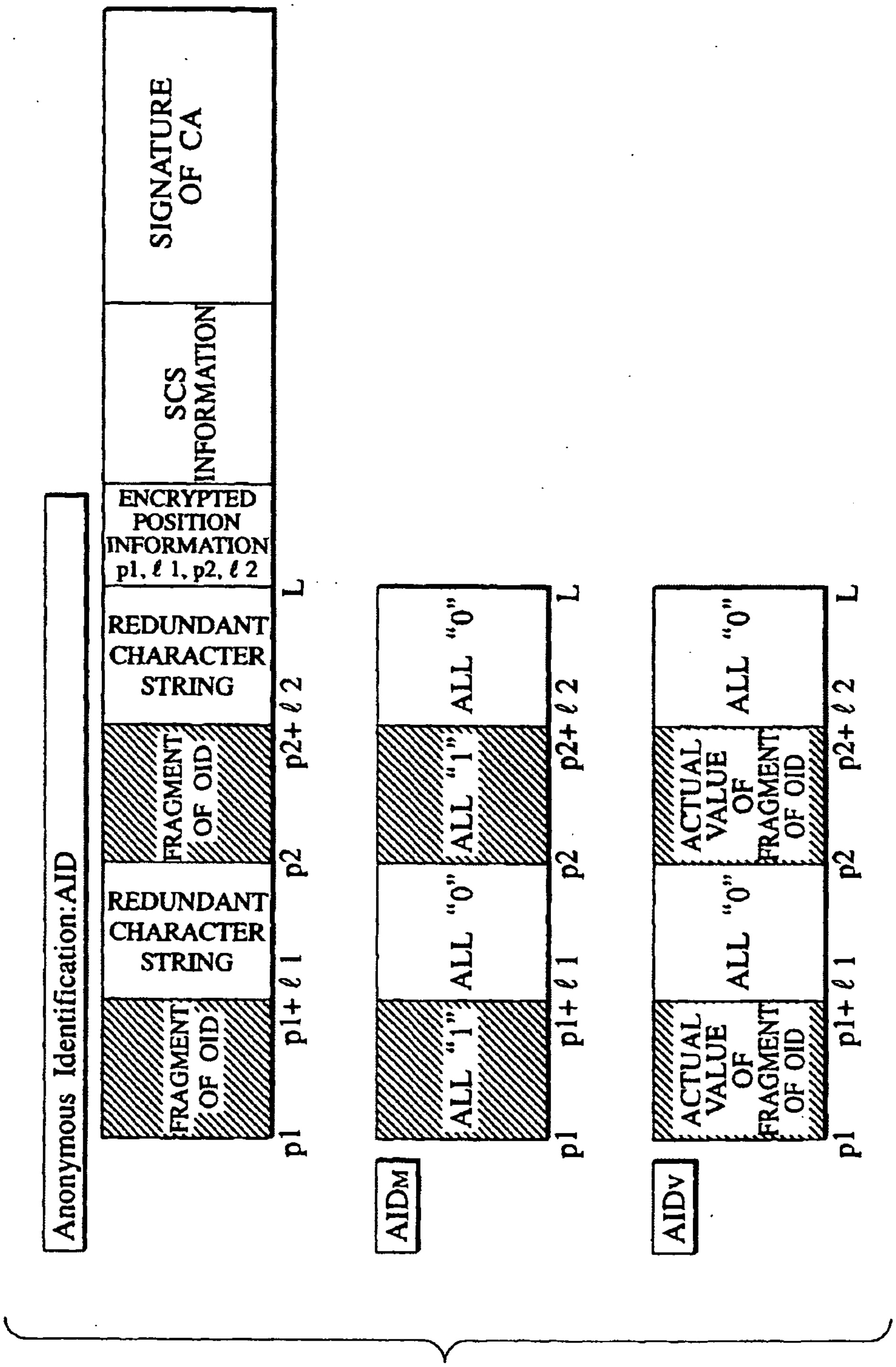

IDENTITY JUDGEMENT
OIDM=0,OIDV=0
S2511
ENTER AID
S2513
GENERATE AIDM & AIDV
S2515
OVRM=OIDM & AIDM
S2517
S2519
OIDV & OVRM=AIDV & OVRM?
NO
YES
OIDM | =AIDM
S2521
OIDV | =AIDV
S2523
S2525
ANY OTHER AID
TO BE PROCESSED?
YES
NO
OUTPUT OIDM & OIDV
S2527
END

FIG.7



Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
ENCRYPTED POSITION INFORMATION p1, ℓ 1, p2, ℓ 2
SCS INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
p2
p2+ ℓ 2
L
AIDM
ALL "1"
ALL "0"
ALL "1"
ALL "0"
AIDV
ACTUAL VALUE OF FRAGMENT OF OID
ALL "0"
ACTUAL VALUE OF FRAGMENT OF OID
ALL "0"

FIG.8

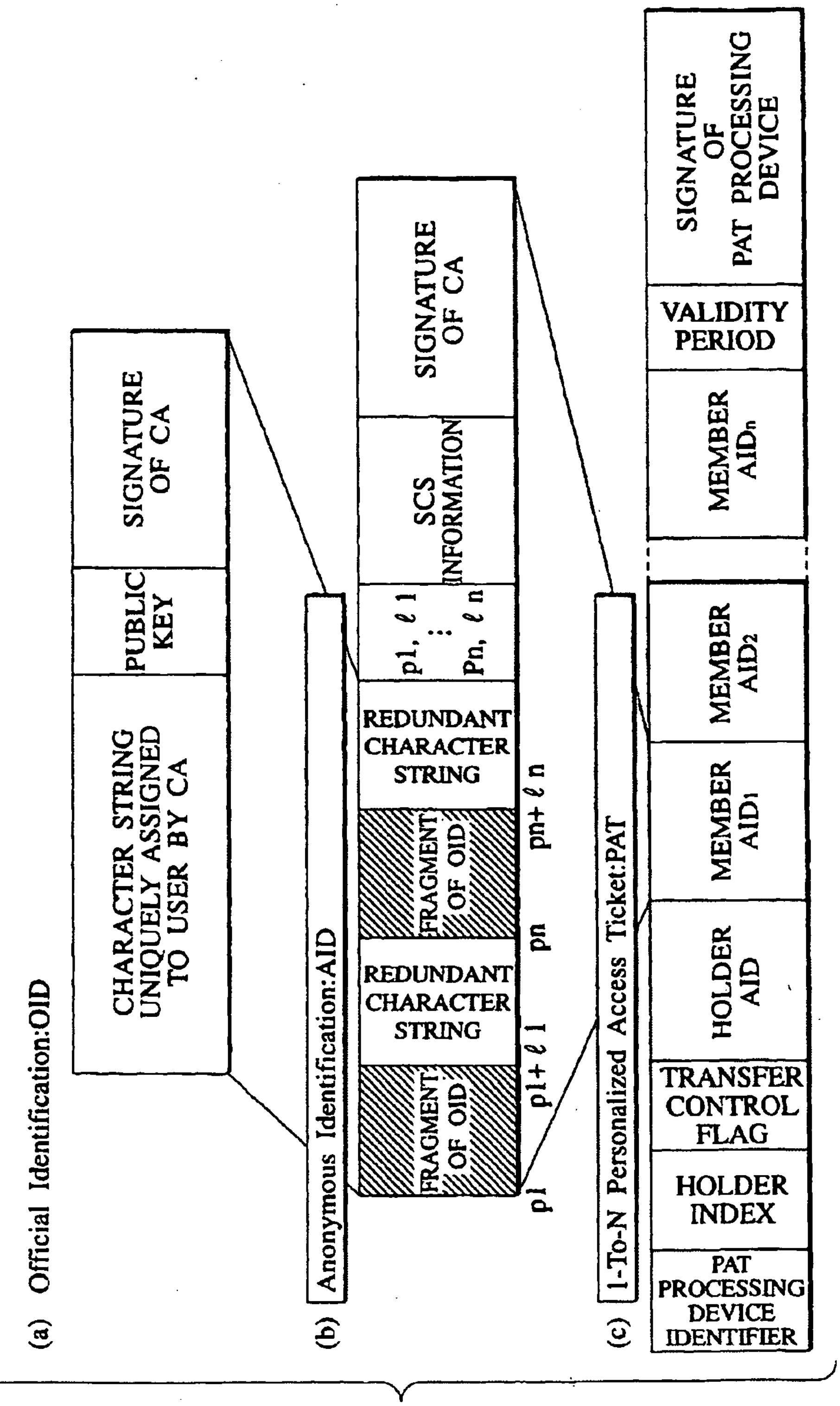

(a) Official Identification:OID
CHARACTER STRING UNIQUELY ASSIGNED TO USER BY CA
PUBLIC KEY
SIGNATURE OF CA
(b) Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
p1, ℓ 1
Pn, ℓ n
SCS INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
pn
pn+ ℓ n
(c) 1-To-N Personalized Access Ticket:PAT
PAT PROCESSING DEVICE IDENTIFIER
HOLDER INDEX
TRANSFER CONTROL FLAG
HOLDER AID
MEMBER AID1
MEMBER AID2
MEMBER AIDn
VALIDITY PERIOD
SIGNATURE OF PAT PROCESSING DEVICE

FIG.9

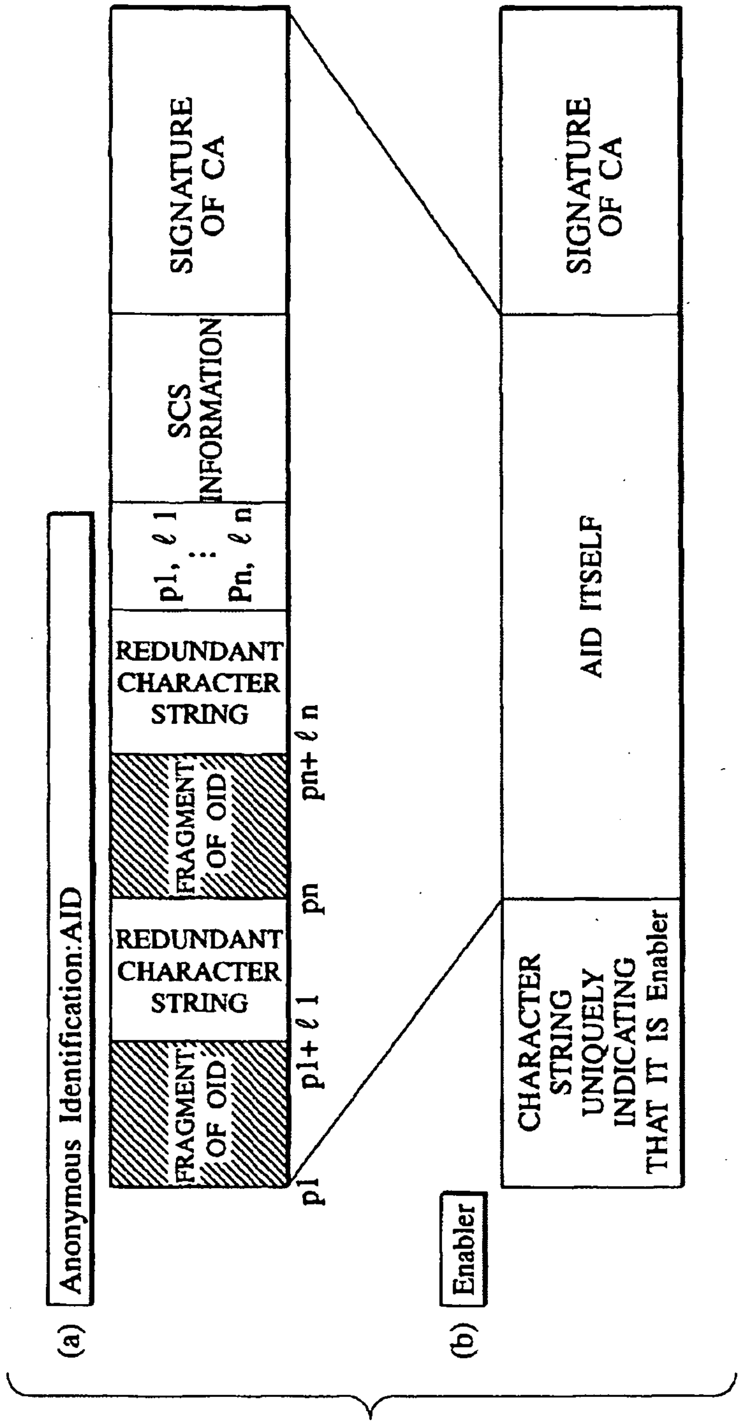

(a)
Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
p1, ℓ 1 … Pn, ℓ n
SCS INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
pn
pn+ ℓ n
(b)
Enabler
CHARACTER STRING UNIQUELY INDICATING THAT IT IS Enabler
AID ITSELF
SIGNATURE OF CA

FIG.10

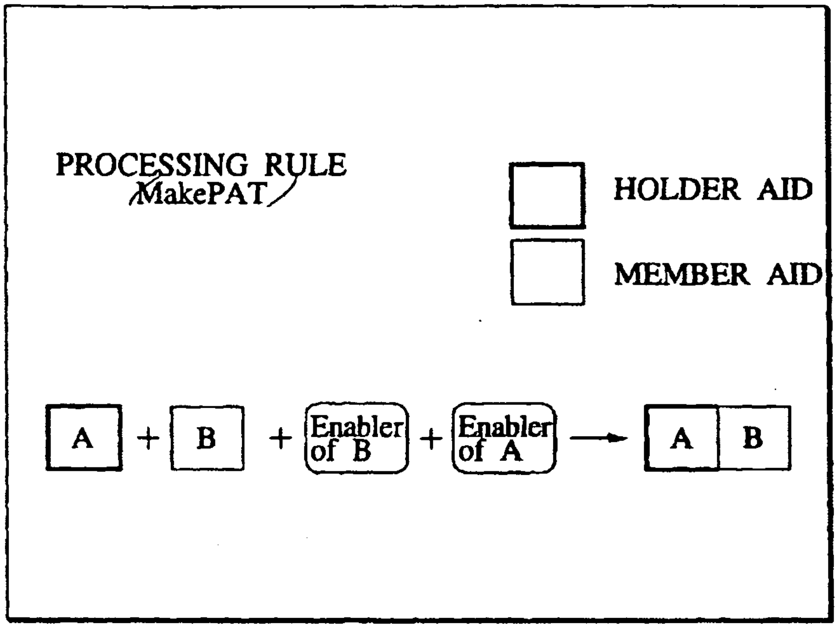

PROCESSING RULE
MakePAT
HOLDER AID
MEMBER AID
A
B
Enabler of B
Enabler of A
A
B

FIG.11

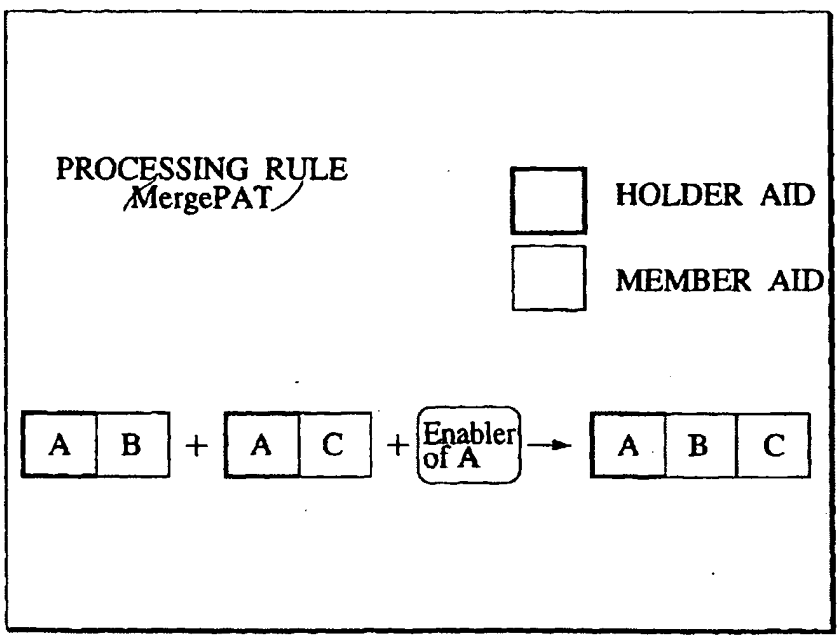

PROCESSING RULE
MergePAT
HOLDER AID
MEMBER AID
A
B
A
C
Enabler of A
A
B
C

FIG.12

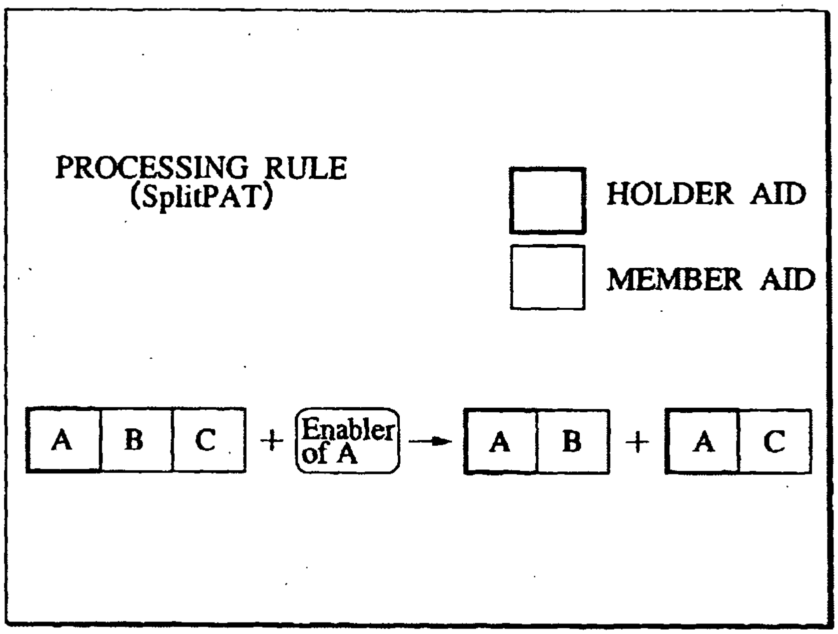

PROCESSING RULE
(SplitPAT)
HOLDER AID
MEMBER AID
A
B
C
+
Enabler of A
A
B
+
A
C

FIG.13

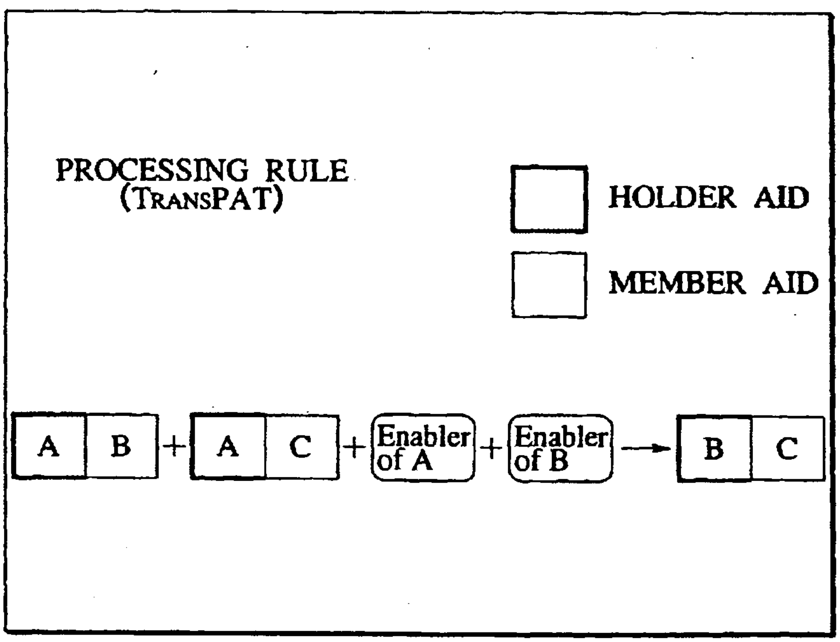
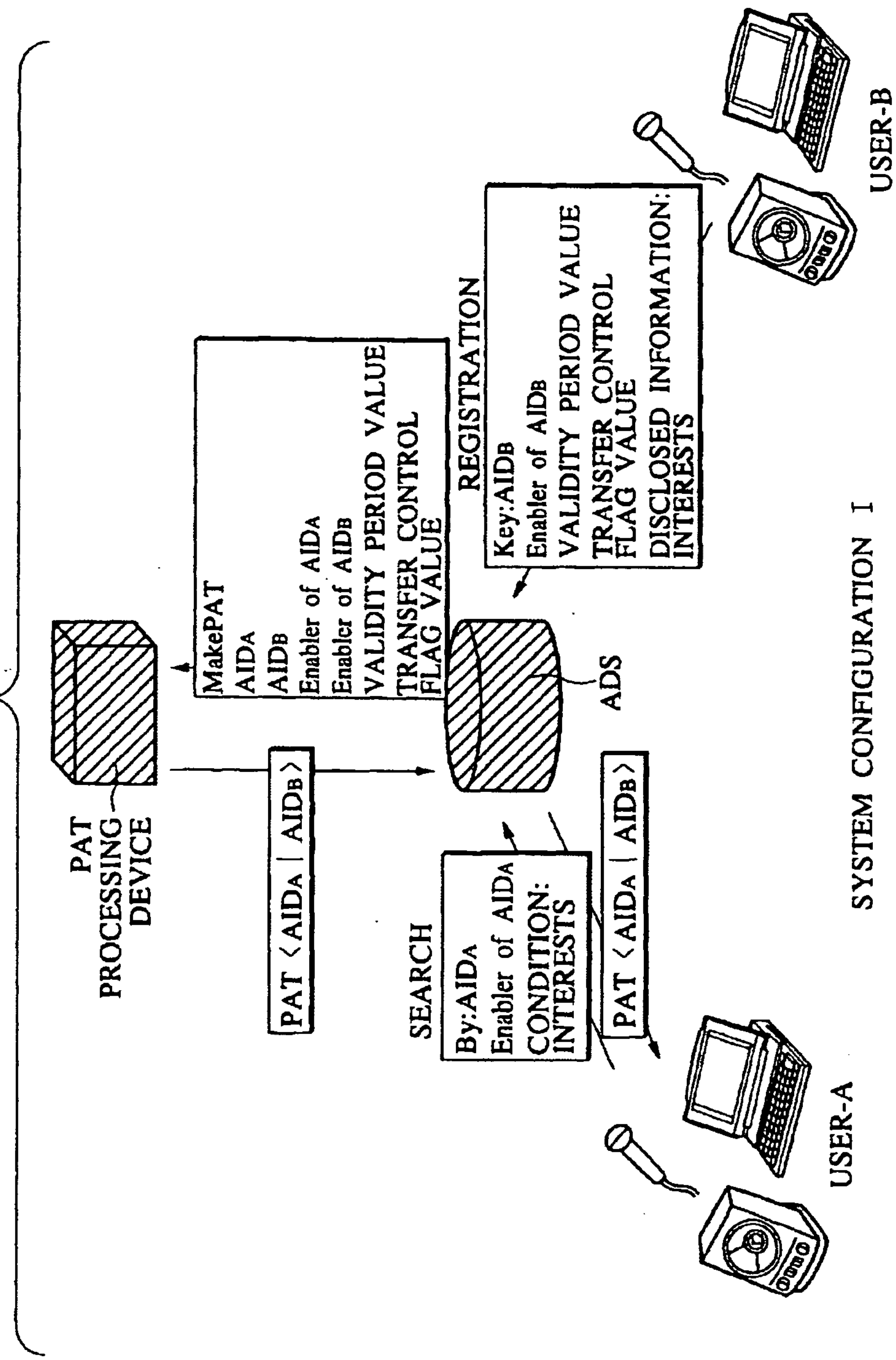
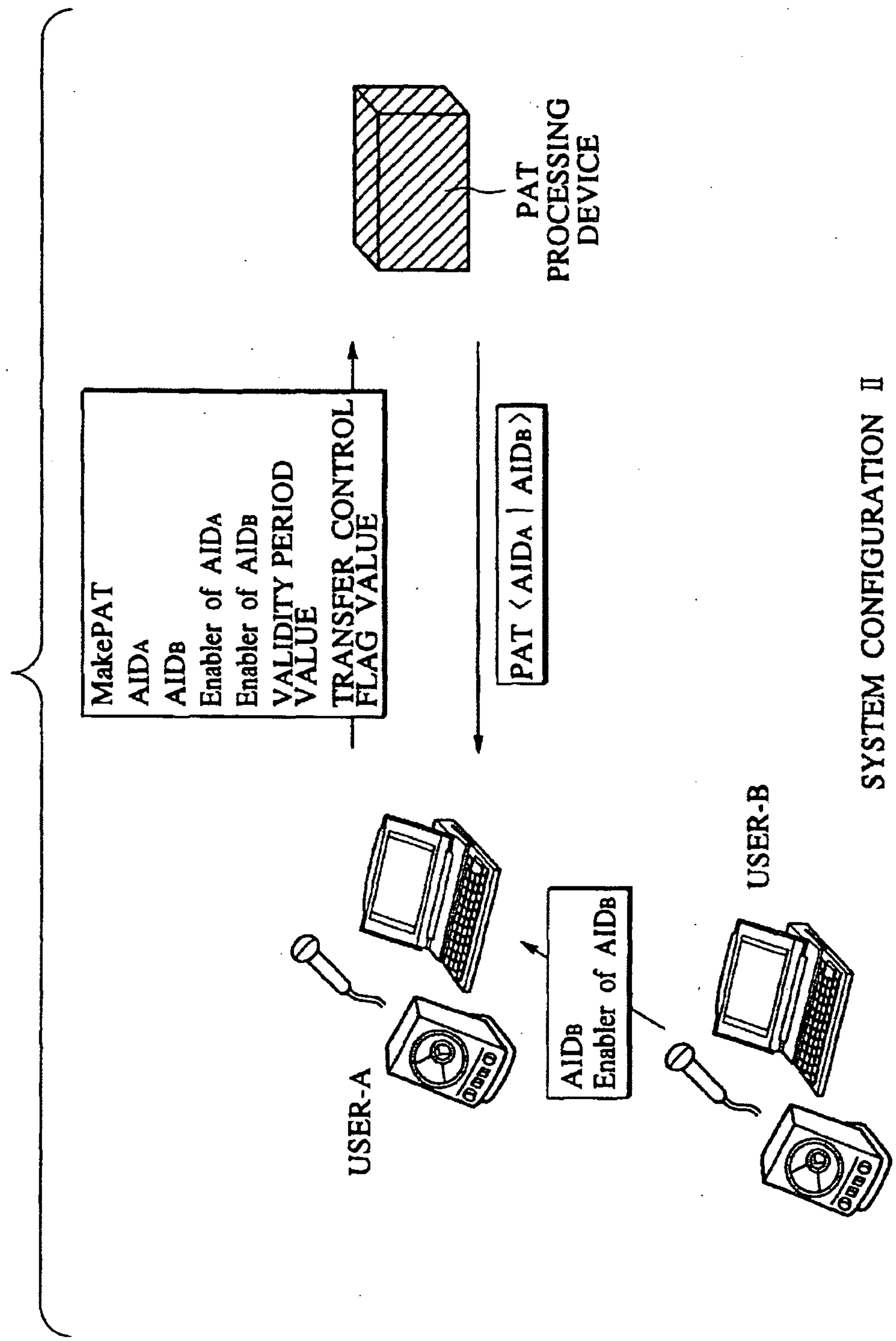

PROCESSING RULE
(TRANSPAT)
HOLDER AID
MEMBER AID
A
B
+
A
C
+
Enabler of A
+
Enabler of B
B
C

FIG.14



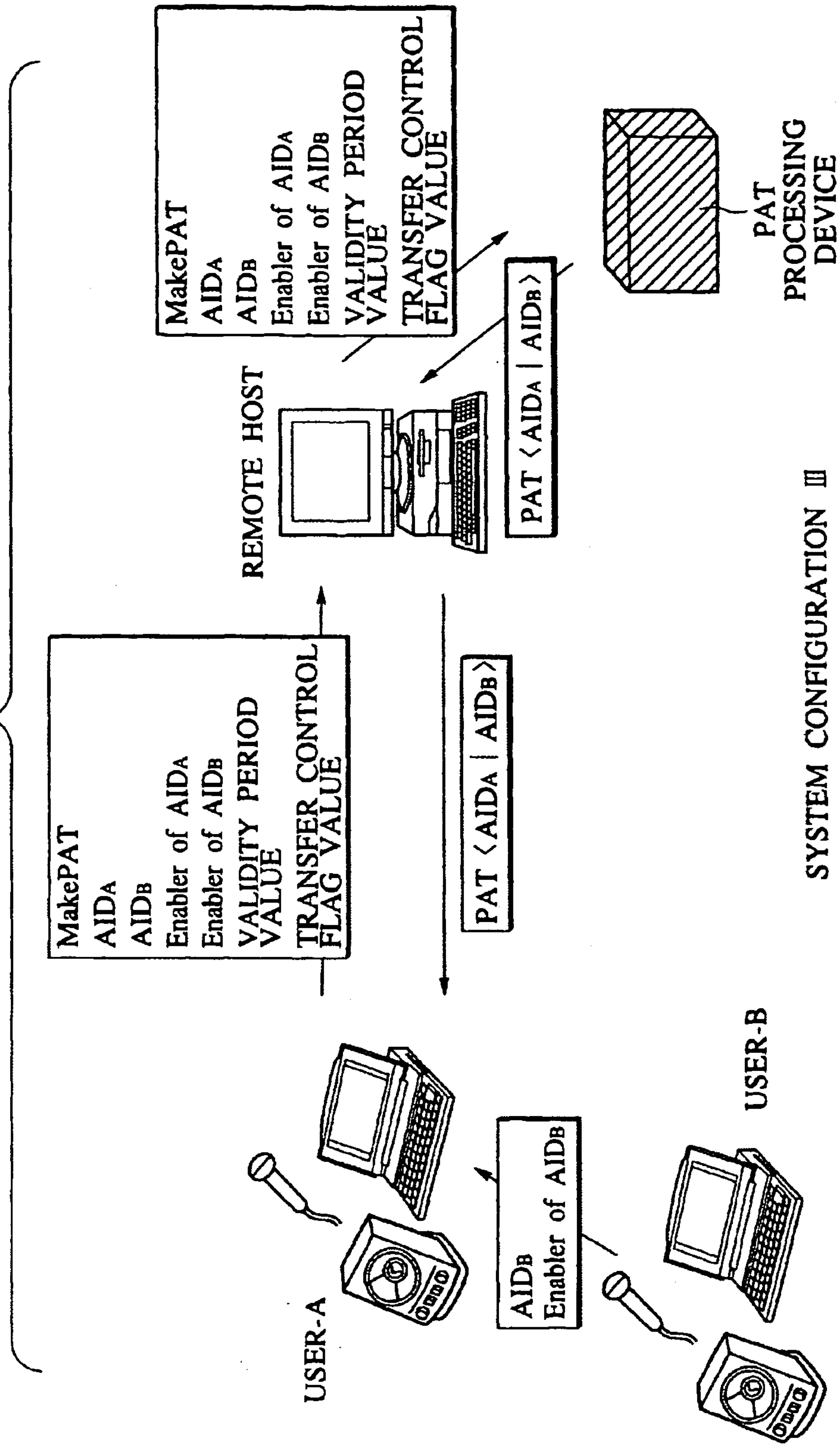
SYSTEM CONFIGURATION I

FIG.15



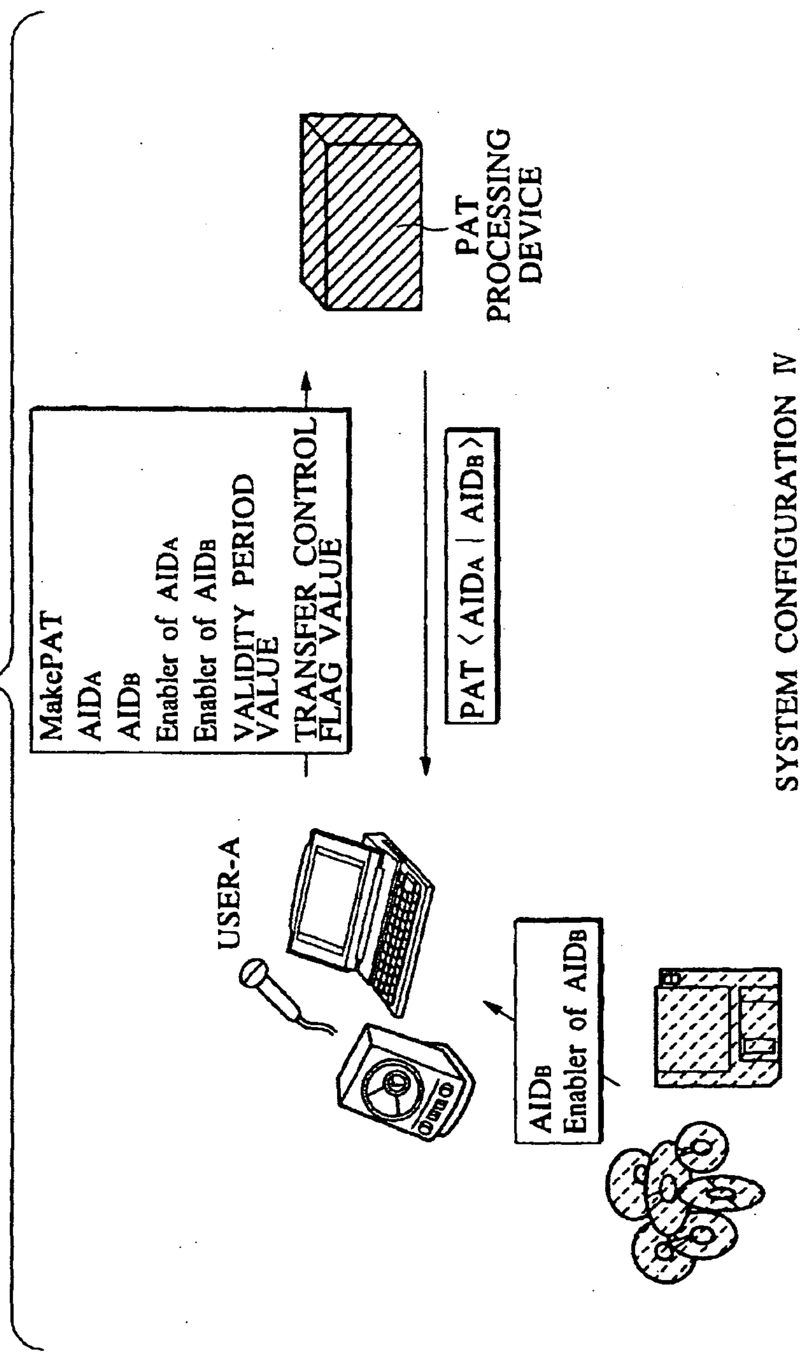
SYSTEM CONFIGURATION II

FIG.16



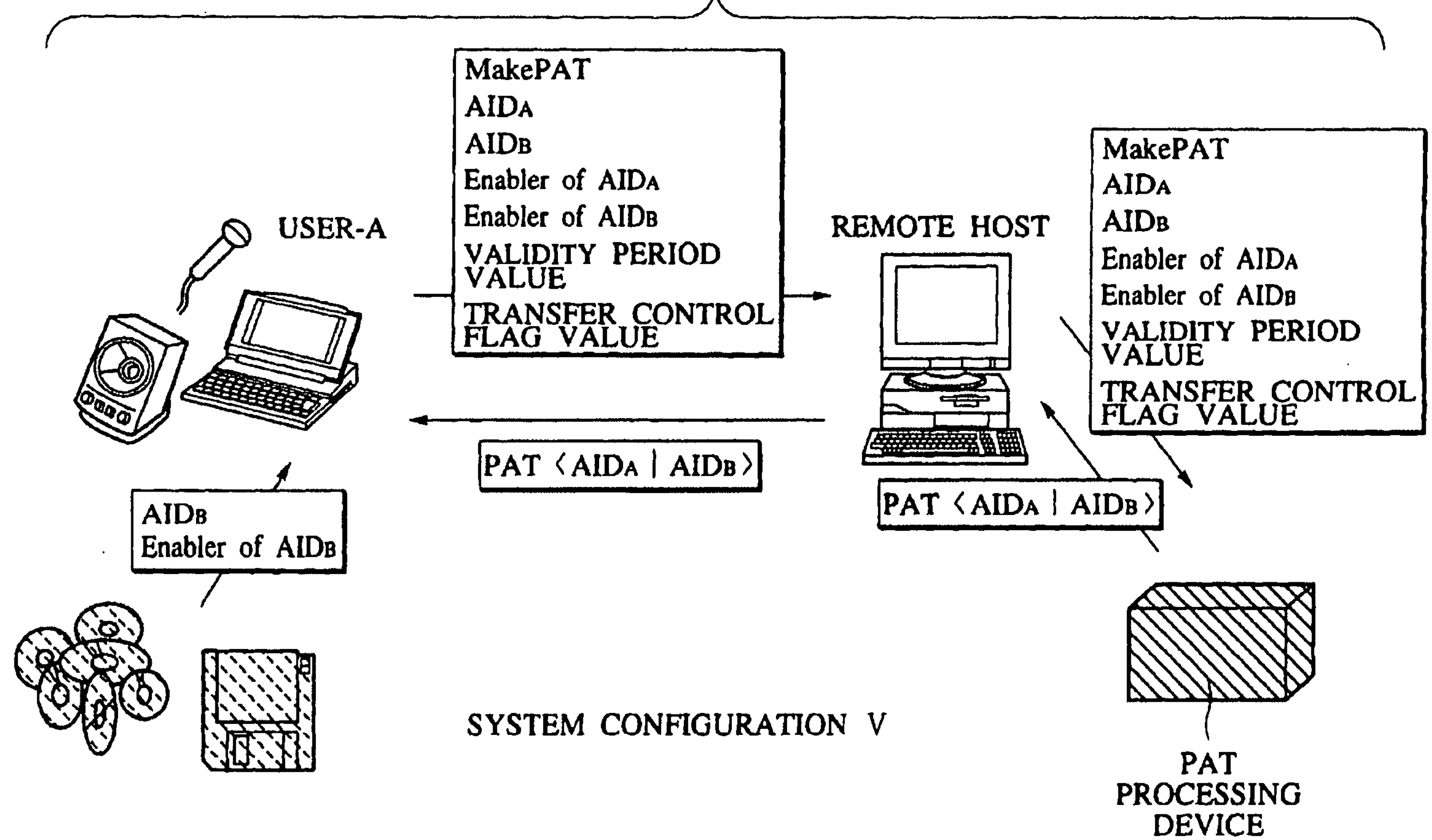
SYSTEM CONFIGURATION III

FIG.17



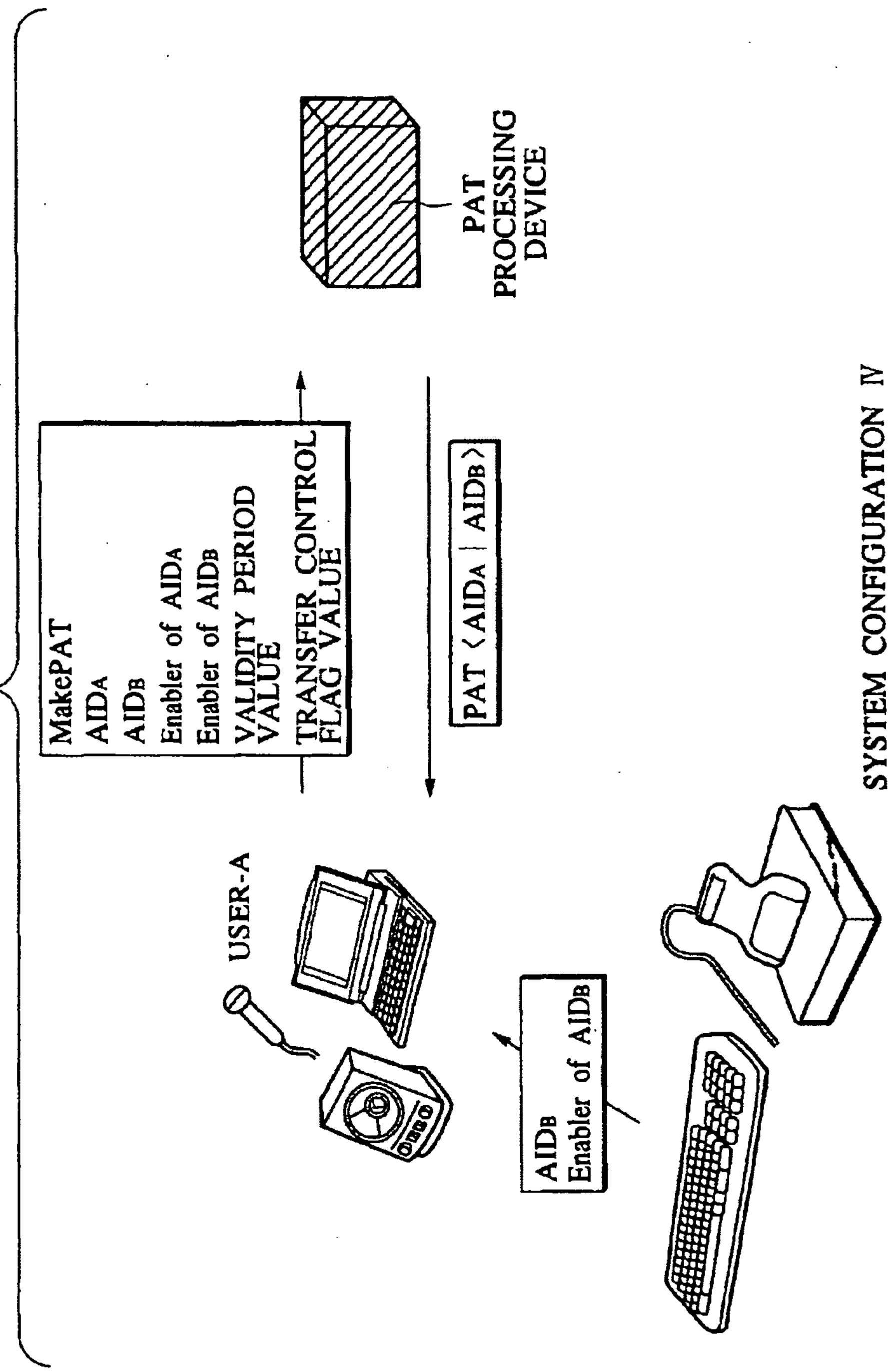
MakePAT
AIDA
AIDB
Enabler of AIDA
Enabler of AIDB
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
PAT ⟨AIDA | AIDB⟩
PAT PROCESSING DEVICE
USER-A
AIDB
Enabler of AIDB
SYSTEM CONFIGURATION IV

FIG.18



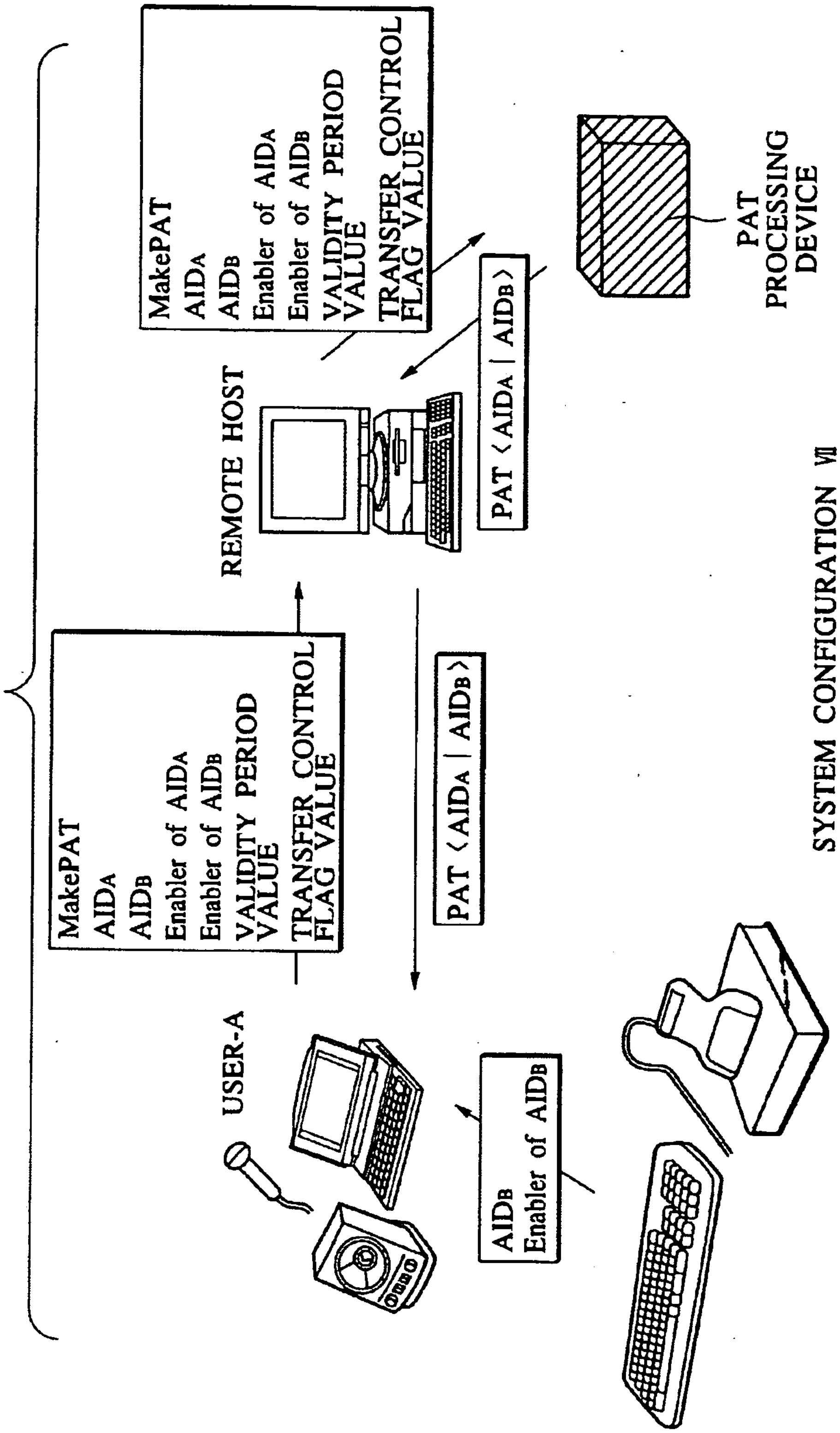
SYSTEM CONFIGURATION V

FIG.19



SYSTEM CONFIGURATION IV

FIG.20



SYSTEM CONFIGURATION VII

FIG.21

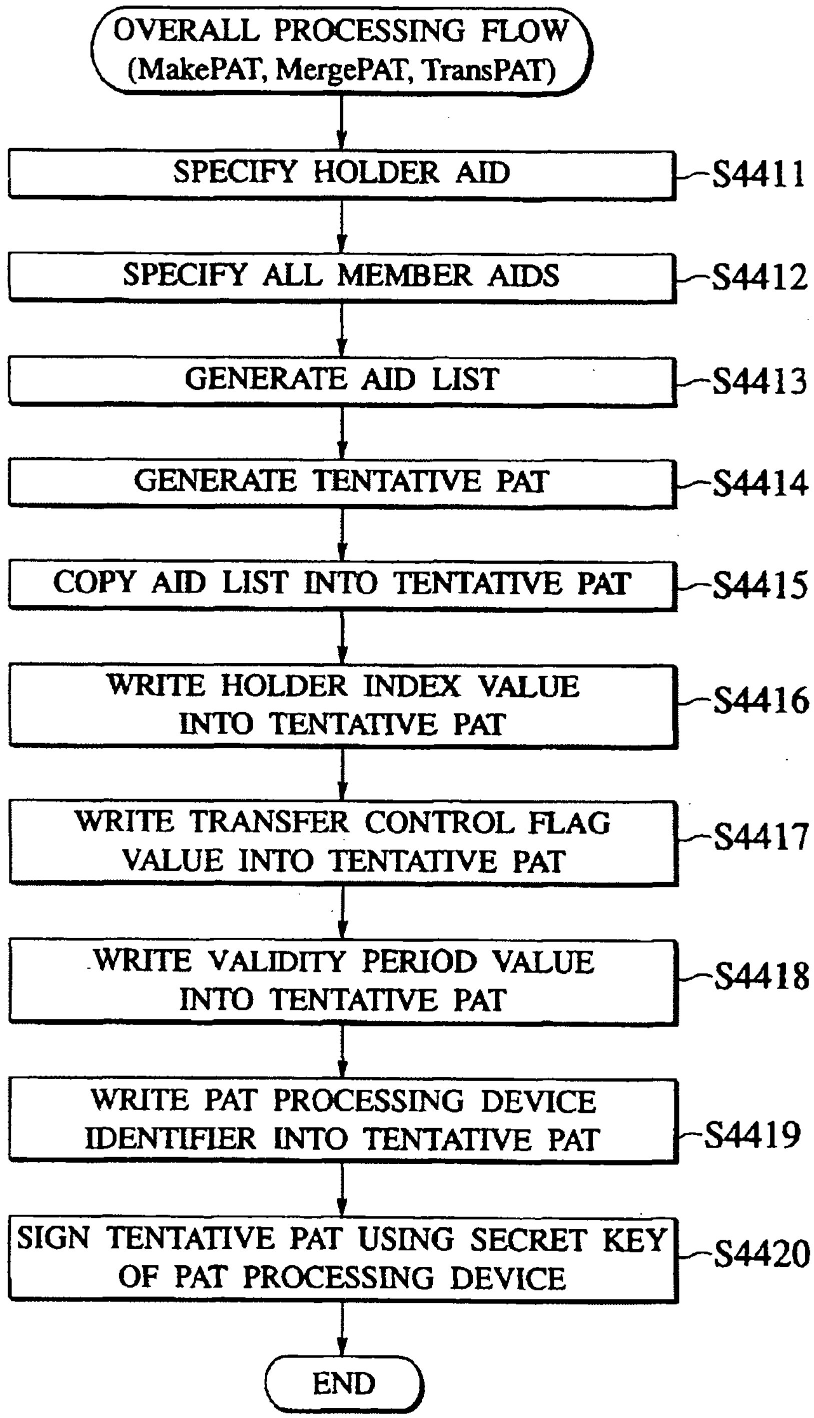

OVERALL PROCESSING FLOW
(MakePAT, MergePAT, TransPAT)
SPECIFY HOLDER AID
S4411
SPECIFY ALL MEMBER AIDS
S4412
GENERATE AID LIST
S4413
GENERATE TENTATIVE PAT
S4414
COPY AID LIST INTO TENTATIVE PAT
S4415
WRITE HOLDER INDEX VALUE INTO TENTATIVE PAT
S4416
WRITE TRANSFER CONTROL FLAG VALUE INTO TENTATIVE PAT
S4417
WRITE VALIDITY PERIOD VALUE INTO TENTATIVE PAT
S4418
WRITE PAT PROCESSING DEVICE IDENTIFIER INTO TENTATIVE PAT
S4419
SIGN TENTATIVE PAT USING SECRET KEY OF PAT PROCESSING DEVICE
S4420
END

FIG.22

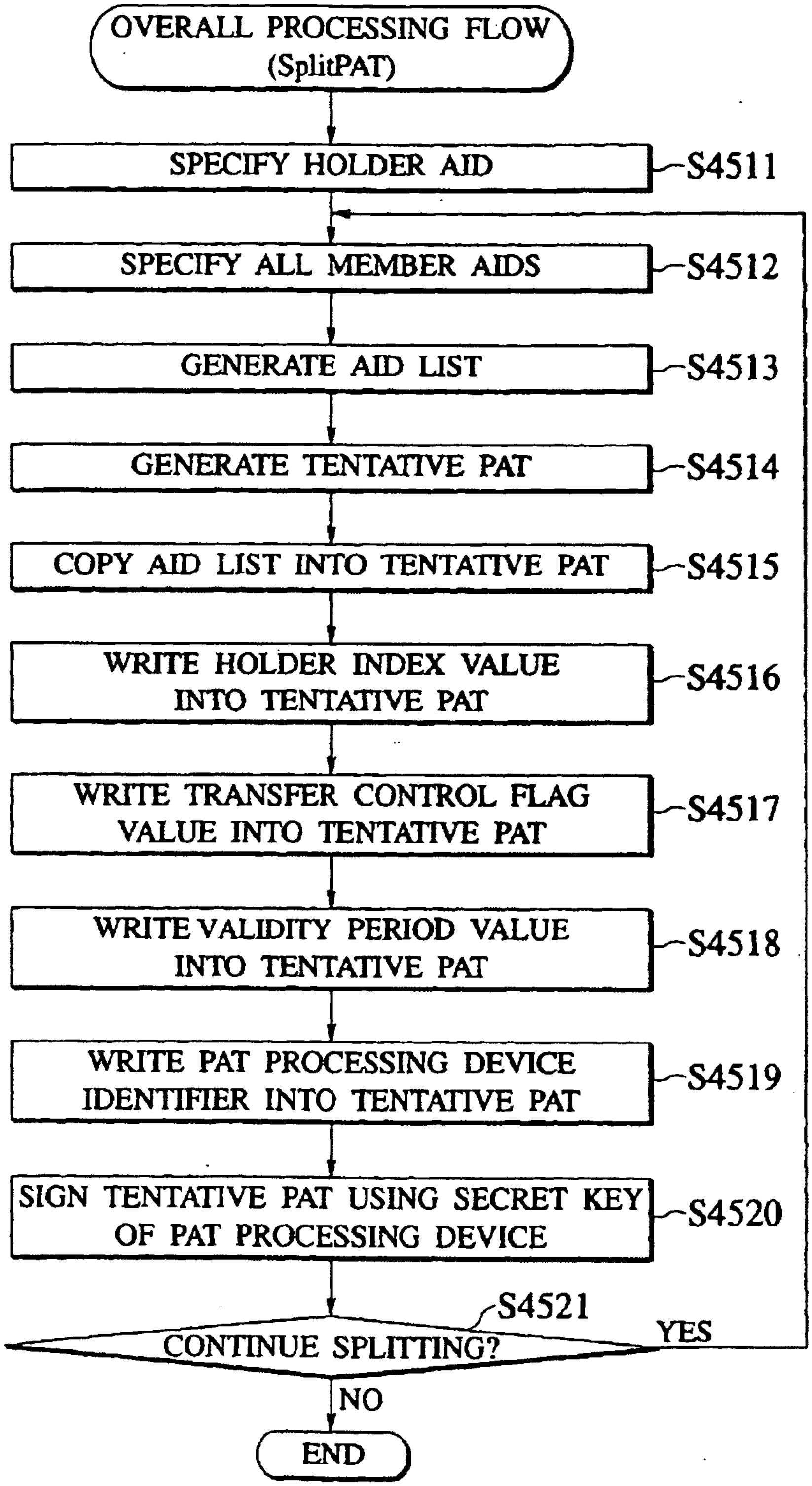

OVERALL PROCESSING FLOW (SplitPAT)
SPECIFY HOLDER AID
S4511
SPECIFY ALL MEMBER AIDS
S4512
GENERATE AID LIST
S4513
GENERATE TENTATIVE PAT
S4514
COPY AID LIST INTO TENTATIVE PAT
S4515
WRITE HOLDER INDEX VALUE INTO TENTATIVE PAT
S4516
WRITE TRANSFER CONTROL FLAG VALUE INTO TENTATIVE PAT
S4517
WRITE VALIDITY PERIOD VALUE INTO TENTATIVE PAT
S4518
WRITE PAT PROCESSING DEVICE IDENTIFIER INTO TENTATIVE PAT
S4519
SIGN TENTATIVE PAT USING SECRET KEY OF PAT PROCESSING DEVICE
S4520
S4521
CONTINUE SPLITTING?
YES
NO
END

FIG.23

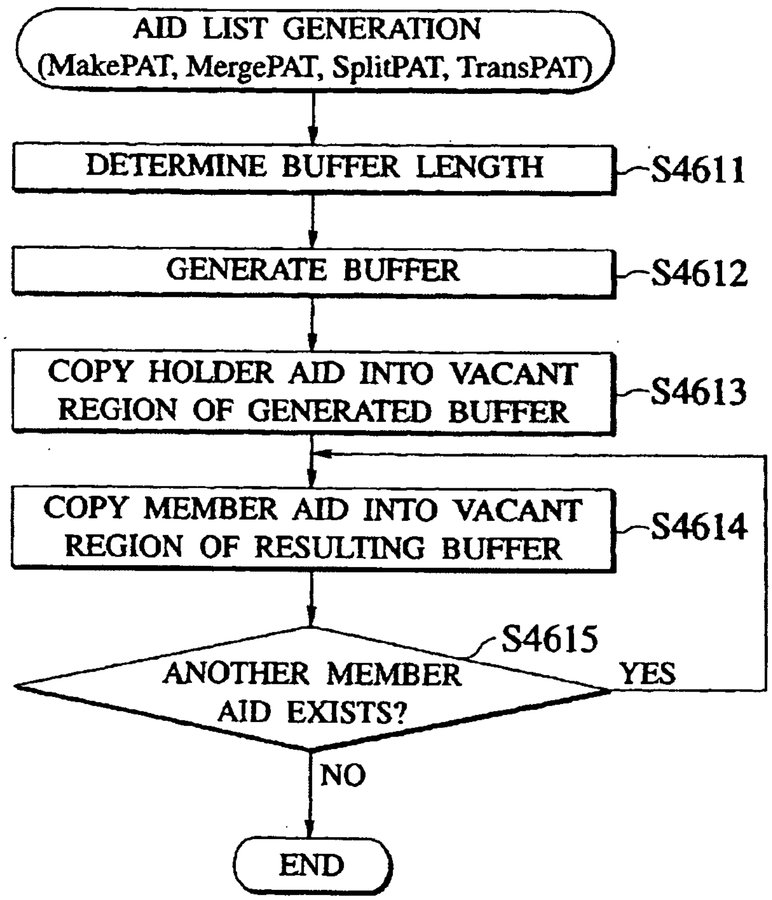

AID LIST GENERATION
(MakePAT, MergePAT, SplitPAT, TransPAT)
DETERMINE BUFFER LENGTH
S4611
GENERATE BUFFER
S4612
COPY HOLDER AID INTO VACANT REGION OF GENERATED BUFFER
S4613
COPY MEMBER AID INTO VACANT REGION OF RESULTING BUFFER
S4614
S4615
ANOTHER MEMBER AID EXISTS?
YES
NO
END

FIG.24

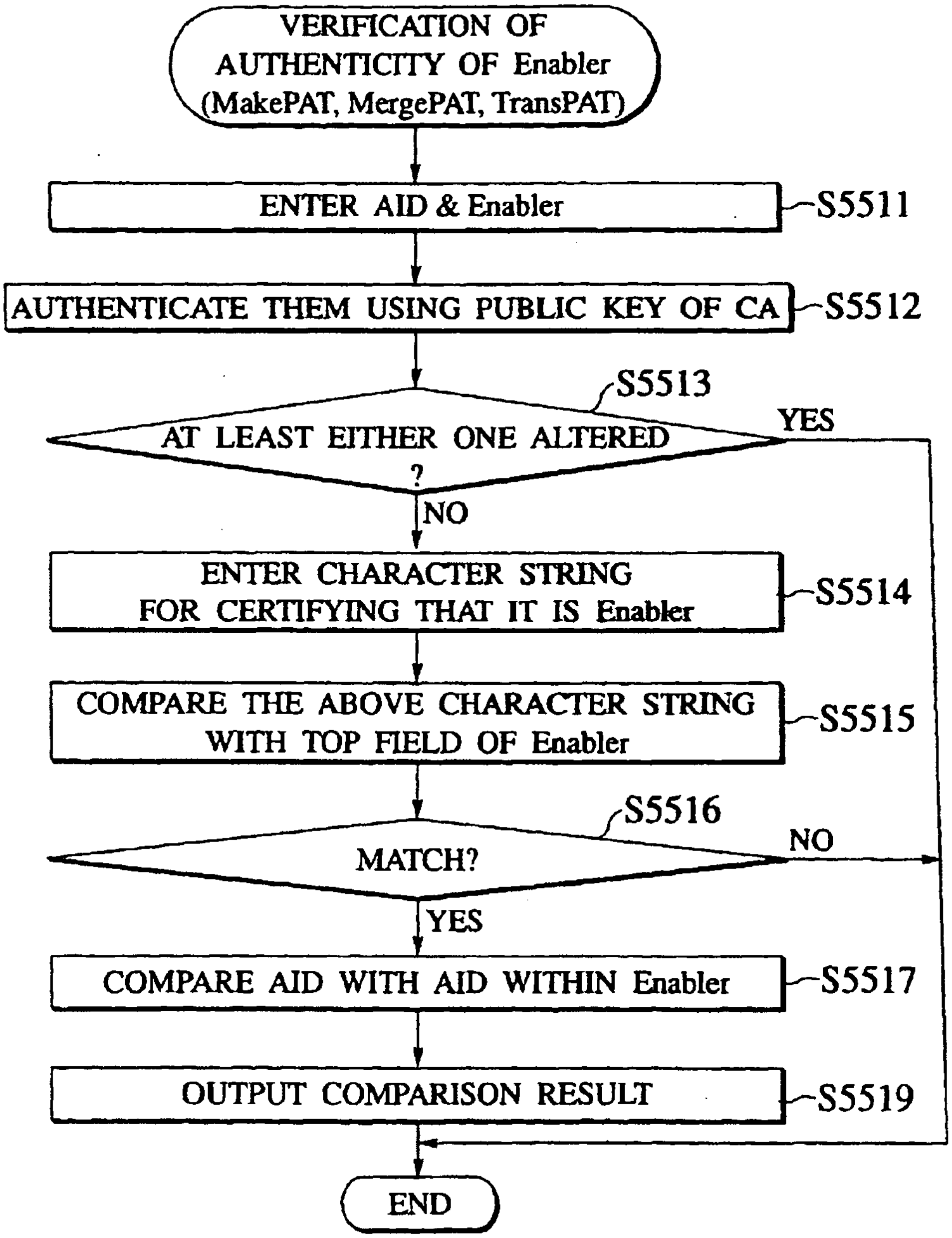

VERIFICATION OF AUTHENTICITY OF Enabler (MakePAT, MergePAT, TransPAT)
ENTER AID & Enabler
S5511
AUTHENTICATE THEM USING PUBLIC KEY OF CA
S5512
S5513
AT LEAST EITHER ONE ALTERED ?
YES
NO
ENTER CHARACTER STRING FOR CERTIFYING THAT IT IS Enabler
S5514
COMPARE THE ABOVE CHARACTER STRING WITH TOP FIELD OF Enabler
S5515
S5516
MATCH?
NO
YES
COMPARE AID WITH AID WITHIN Enabler
S5517
OUTPUT COMPARISON RESULT
S5519
END

FIG.25

DATA STRUCTURE OF Null-AID

CHARACTER STRING UNIQUELY INDICATING THAT IT IS Null-AID	SIGNATURE OF CA

FIG.26

DATA STRUCTURE OF Enabler of Null-AID

CHARACTER STRING UNIQUELY INDICATING THAT IT IS Enabler	Null-AID ITSELF	SIGNATURE OF CA

FIG.27

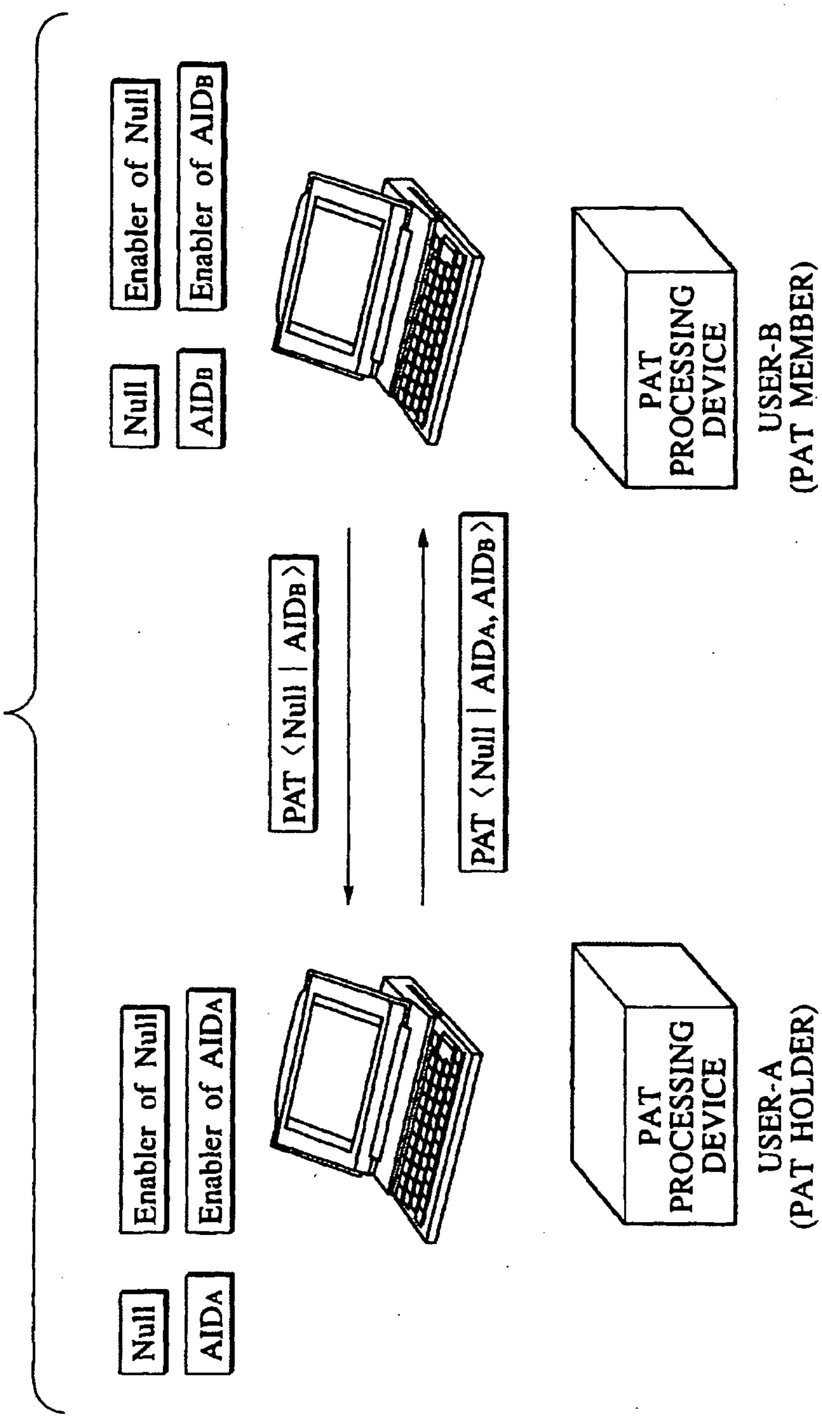

Null
Enabler of Null
AID$_A$
Enabler of AID$_A$
PAT PROCESSING DEVICE
USER-A
(PAT HOLDER)
PAT ⟨Null | AID$_B$⟩
PAT ⟨Null | AID$_A$, AID$_B$⟩
Null
Enabler of Null
AID$_B$
Enabler of AID$_B$
PAT PROCESSING DEVICE
USER-B
(PAT MEMBER)

FIG.28

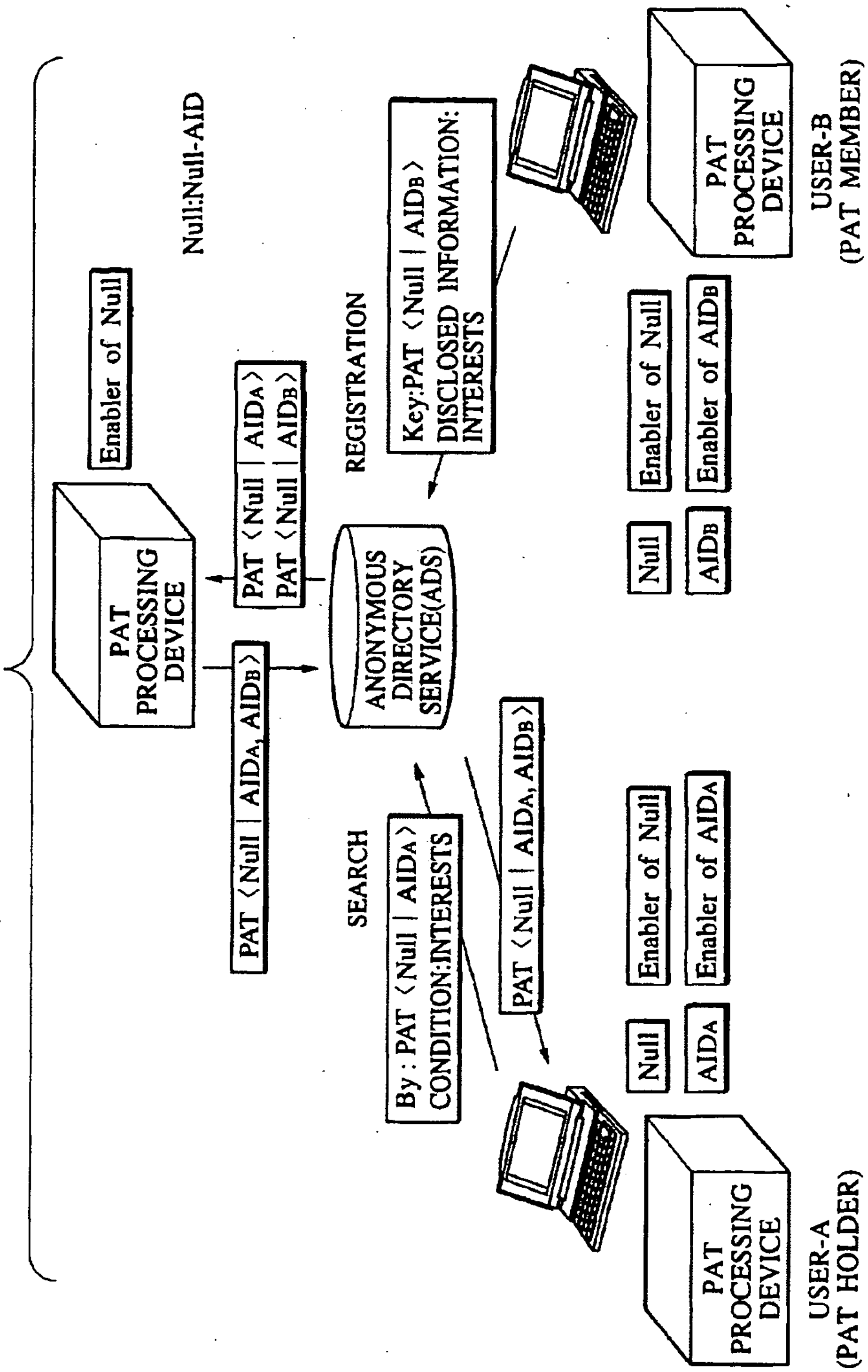

Null:Null-AID
Enabler of Null
PAT PROCESSING DEVICE
PAT ⟨Null | AIDA⟩
PAT ⟨Null | AIDB⟩
PAT ⟨Null | AIDA, AIDB⟩
ANONYMOUS DIRECTORY SERVICE(ADS)
REGISTRATION
Key:PAT ⟨Null | AIDB⟩
DISCLOSED INFORMATION:
INTERESTS
SEARCH
By : PAT ⟨Null | AIDA⟩
CONDITION:INTERESTS
PAT ⟨Null | AIDA, AIDB⟩
Null
Enabler of Null
AIDB
Enabler of AIDB
Null
Enabler of Null
AIDA
Enabler of AIDA
PAT PROCESSING DEVICE
USER-B
(PAT MEMBER)
PAT PROCESSING DEVICE
USER-A
(PAT HOLDER)

FIG.29

DATA STRUCTURE OF God-AID

CHARACTER STRING UNIQUELY INDICATING THAT IT IS God-AID	SIGNATURE OF CA

FIG.30

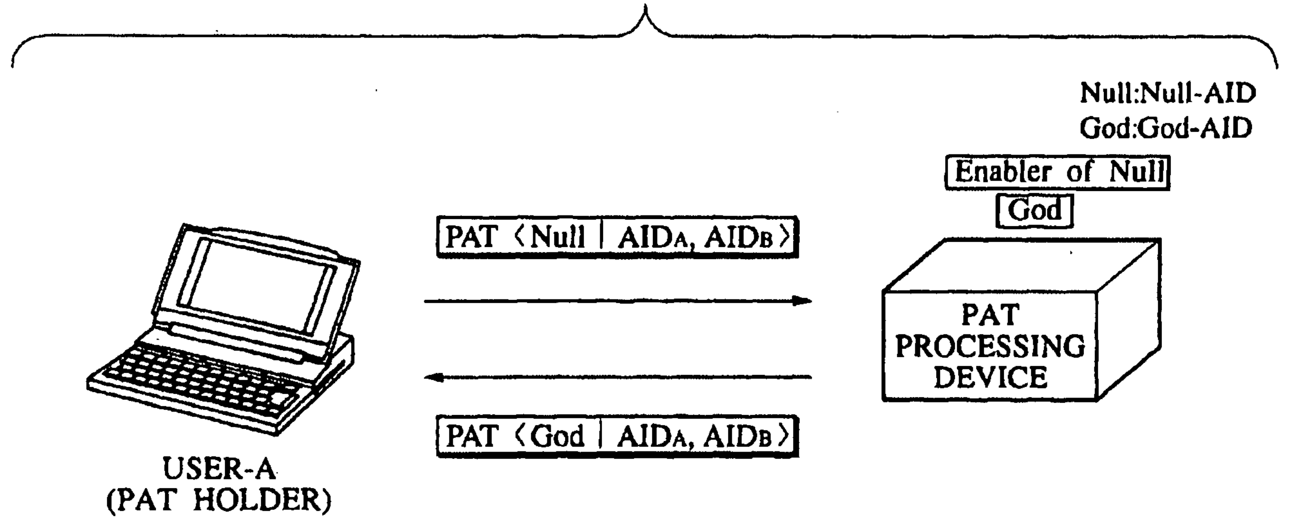


FIG.31

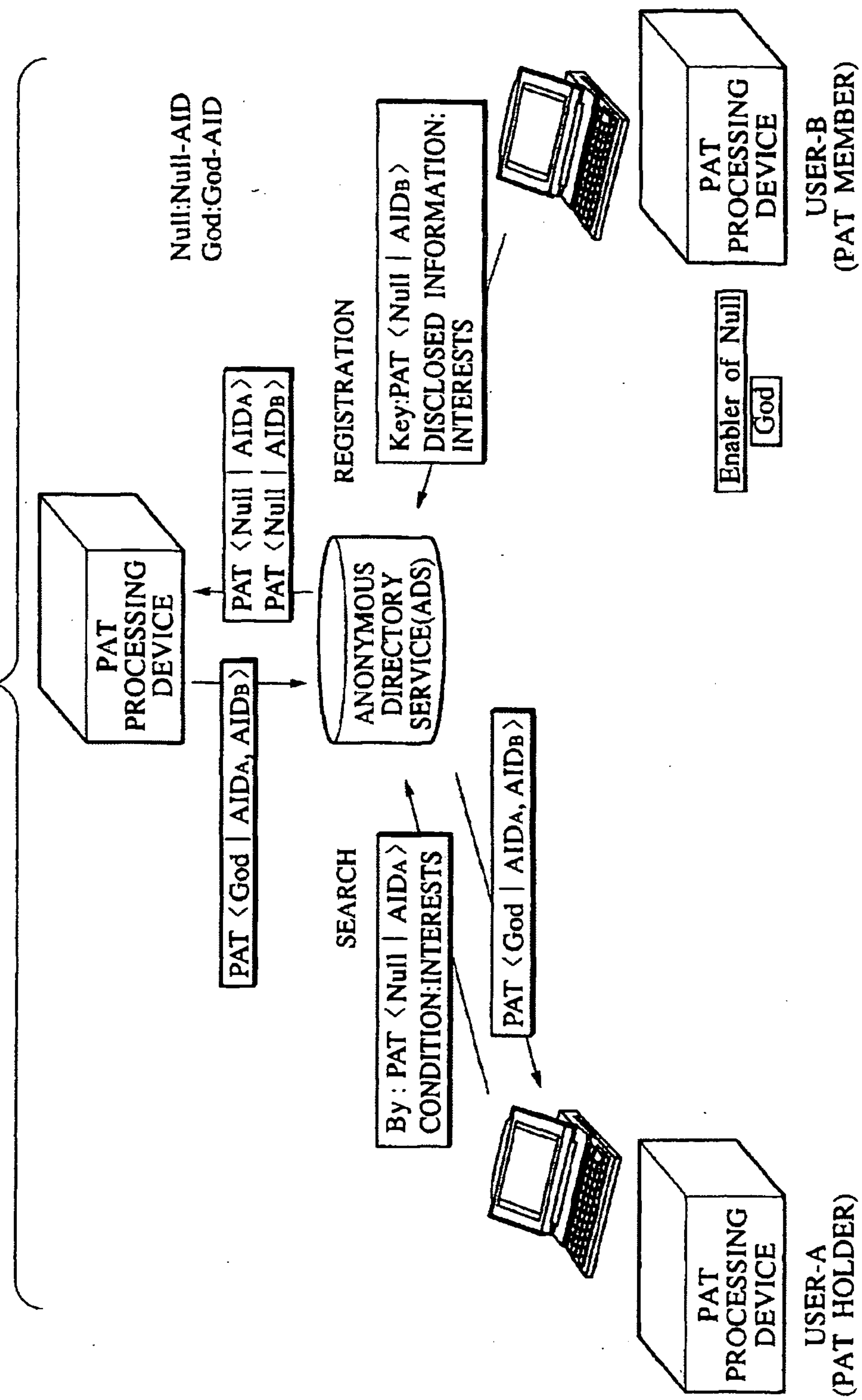

Null:Null-AID
God:God-AID
PAT PROCESSING DEVICE
PAT 〈Null | AIDA〉
PAT 〈Null | AIDB〉
PAT 〈God | AIDA, AIDB〉
ANONYMOUS DIRECTORY SERVICE(ADS)
REGISTRATION
Key:PAT 〈Null | AIDB〉
DISCLOSED INFORMATION:
INTERESTS
SEARCH
By : PAT 〈Null | AIDA〉
CONDITION:INTERESTS
PAT 〈God | AIDA, AIDB〉
Enabler of Null
God
PAT PROCESSING DEVICE
USER-B
(PAT MEMBER)
PAT PROCESSING DEVICE
USER-A
(PAT HOLDER)

FIG.32

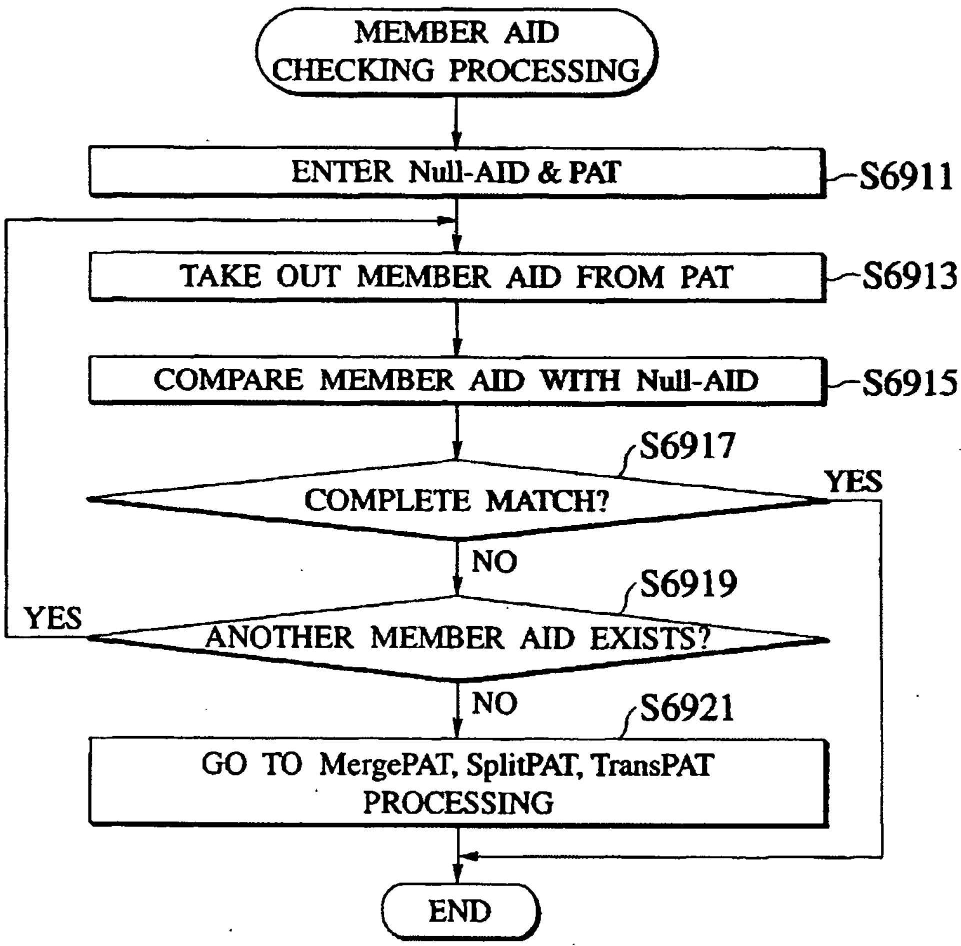

MEMBER AID CHECKING PROCESSING
ENTER Null-AID & PAT
S6911
TAKE OUT MEMBER AID FROM PAT
S6913
COMPARE MEMBER AID WITH Null-AID
S6915
S6917
COMPLETE MATCH?
YES
NO
S6919
YES
ANOTHER MEMBER AID EXISTS?
NO
S6921
GO TO MergePAT, SplitPAT, TransPAT PROCESSING
END

FIG.33

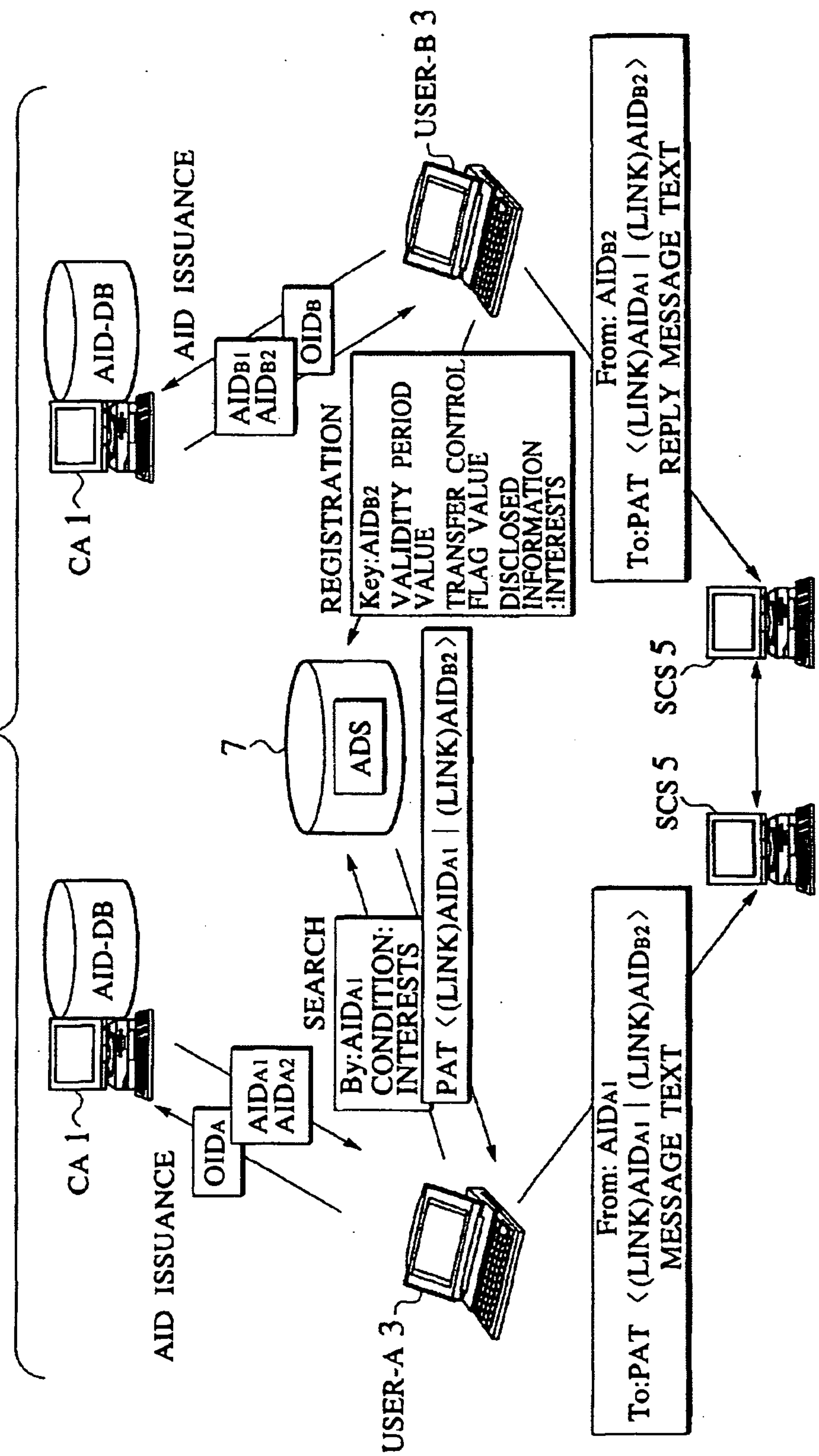

CA 1
AID-DB
AID ISSUANCE
OIDA
AIDA1
AIDA2
CA 1
AID-DB
AID ISSUANCE
AIDB1
AIDB2
OIDB
USER-B 3
REGISTRATION
Key:AIDB2
VALIDITY PERIOD VALUE
TRANSFER CONTROL FLAG VALUE
DISCLOSED INFORMATION :INTERESTS
7
ADS
SEARCH
By:AIDA1
CONDITION: INTERESTS
PAT 〈(LINK)AIDA1 | (LINK)AIDB2〉
USER-A 3
From: AIDB2
To:PAT 〈(LINK)AIDA1 | (LINK)AIDB2〉
REPLY MESSAGE TEXT
SCS 5
SMTP
SCS 5
SMTP
From: AIDA1
To:PAT 〈(LINK)AIDA1 | (LINK)AIDB2〉
MESSAGE TEXT

FIG.34

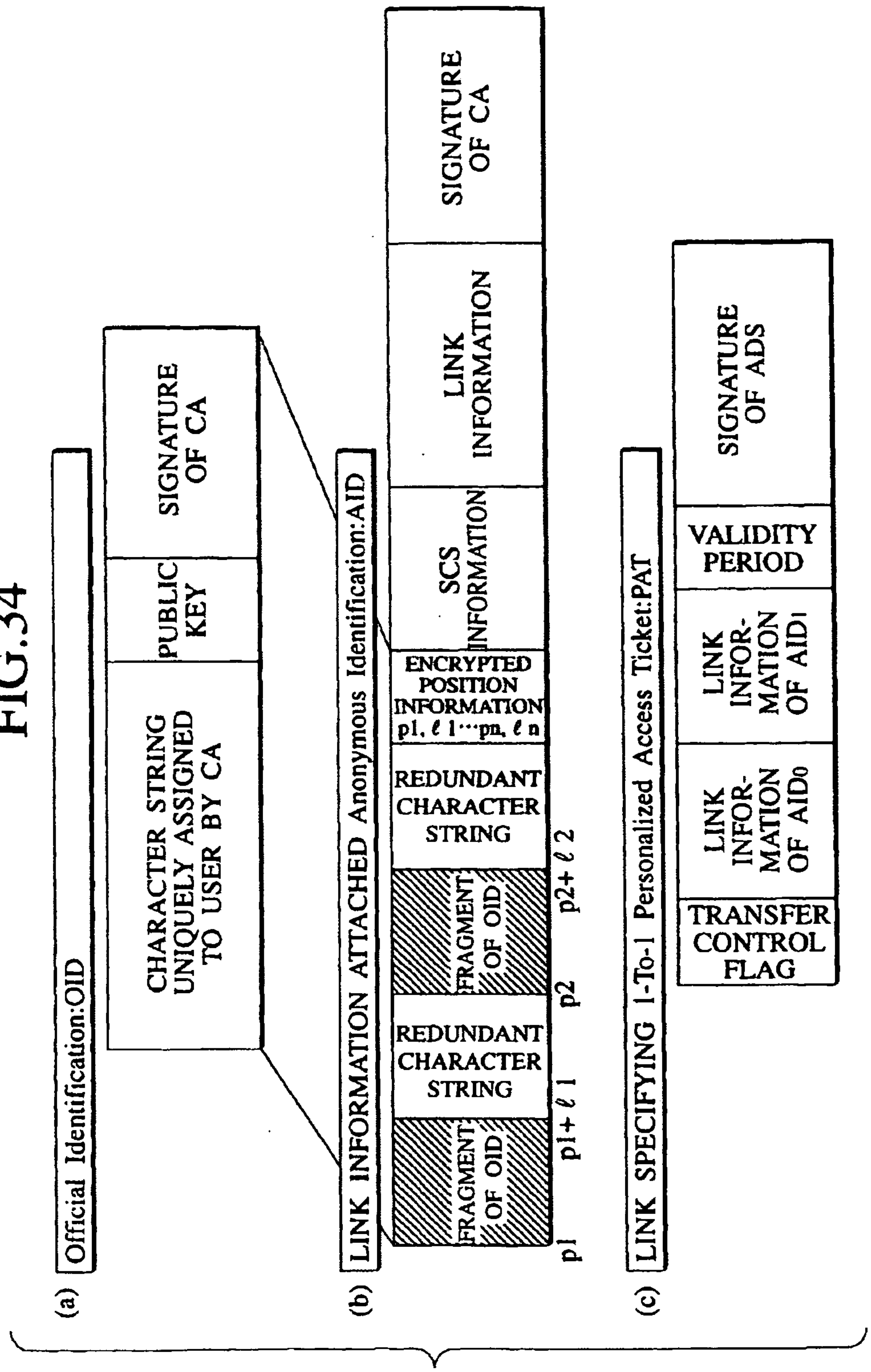

(a) Official Identification:OID
CHARACTER STRING UNIQUELY ASSIGNED TO USER BY CA
PUBLIC KEY
SIGNATURE OF CA
(b) LINK INFORMATION ATTACHED Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
ENCRYPTED POSITION INFORMATION p1, ℓ 1···pn, ℓ n
SCS INFORMATION
LINK INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
p2
p2+ ℓ 2
(c) LINK SPECIFYING 1-To-1 Personalized Access Ticket:PAT
TRANSFER CONTROL FLAG
LINK INFOR-MATION OF AID0
LINK INFOR-MATION OF AID1
VALIDITY PERIOD
SIGNATURE OF ADS

FIG.35

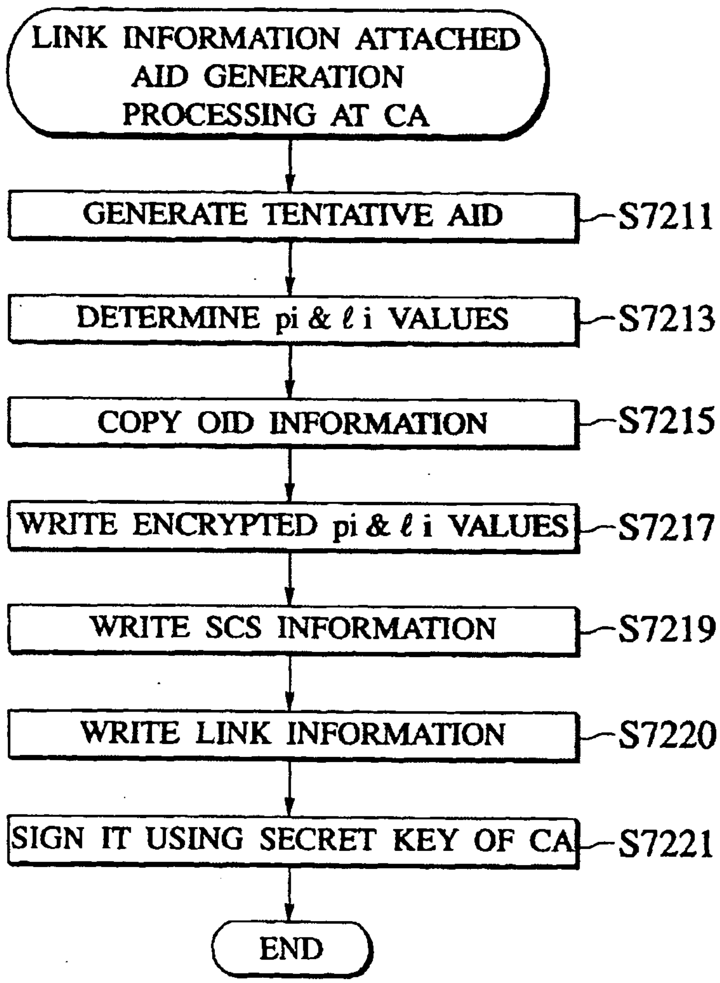

LINK INFORMATION ATTACHED AID GENERATION PROCESSING AT CA
GENERATE TENTATIVE AID
S7211
DETERMINE pi & ℓi VALUES
S7213
COPY OID INFORMATION
S7215
WRITE ENCRYPTED pi & ℓi VALUES
S7217
WRITE SCS INFORMATION
S7219
WRITE LINK INFORMATION
S7220
SIGN IT USING SECRET KEY OF CA
S7221
END

FIG.36

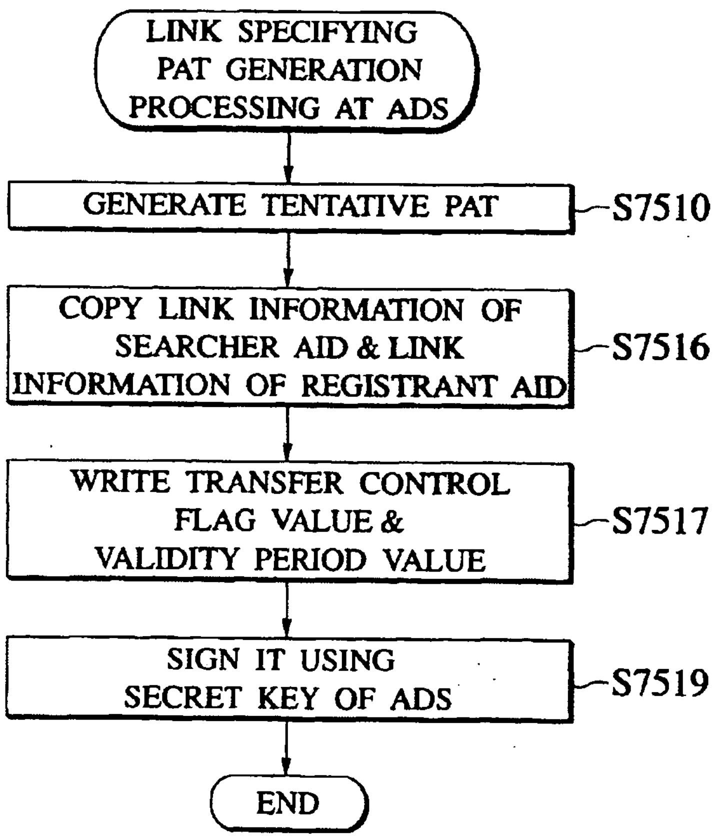

LINK SPECIFYING PAT GENERATION PROCESSING AT ADS
GENERATE TENTATIVE PAT
S7510
COPY LINK INFORMATION OF SEARCHER AID & LINK INFORMATION OF REGISTRANT AID
S7516
WRITE TRANSFER CONTROL FLAG VALUE & VALIDITY PERIOD VALUE
S7517
SIGN IT USING SECRET KEY OF ADS
S7519
END

FIG.37

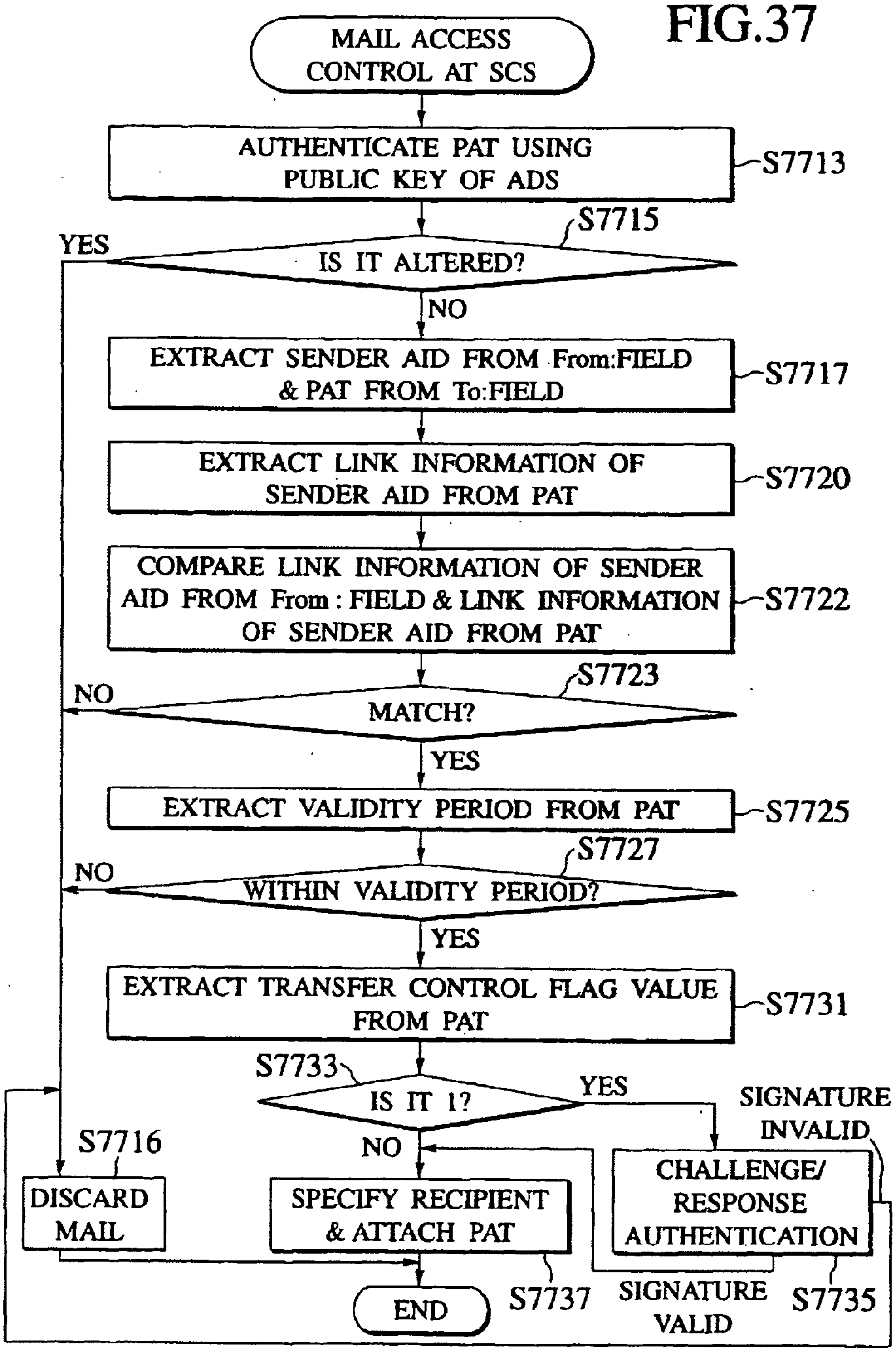

MAIL ACCESS CONTROL AT SCS
AUTHENTICATE PAT USING PUBLIC KEY OF ADS
S7713
S7715
IS IT ALTERED?
YES
NO
EXTRACT SENDER AID FROM From:FIELD & PAT FROM To:FIELD
S7717
EXTRACT LINK INFORMATION OF SENDER AID FROM PAT
S7720
COMPARE LINK INFORMATION OF SENDER AID FROM From : FIELD & LINK INFORMATION OF SENDER AID FROM PAT
S7722
S7723
MATCH?
NO
YES
EXTRACT VALIDITY PERIOD FROM PAT
S7725
S7727
WITHIN VALIDITY PERIOD?
NO
YES
EXTRACT TRANSFER CONTROL FLAG VALUE FROM PAT
S7731
S7733
IS IT 1?
YES
NO
SIGNATURE INVALID
S7716
DISCARD MAIL
SPECIFY RECIPIENT & ATTACH PAT
CHALLENGE/ RESPONSE AUTHENTICATION
END
S7737
SIGNATURE VALID
S7735

FIG.38

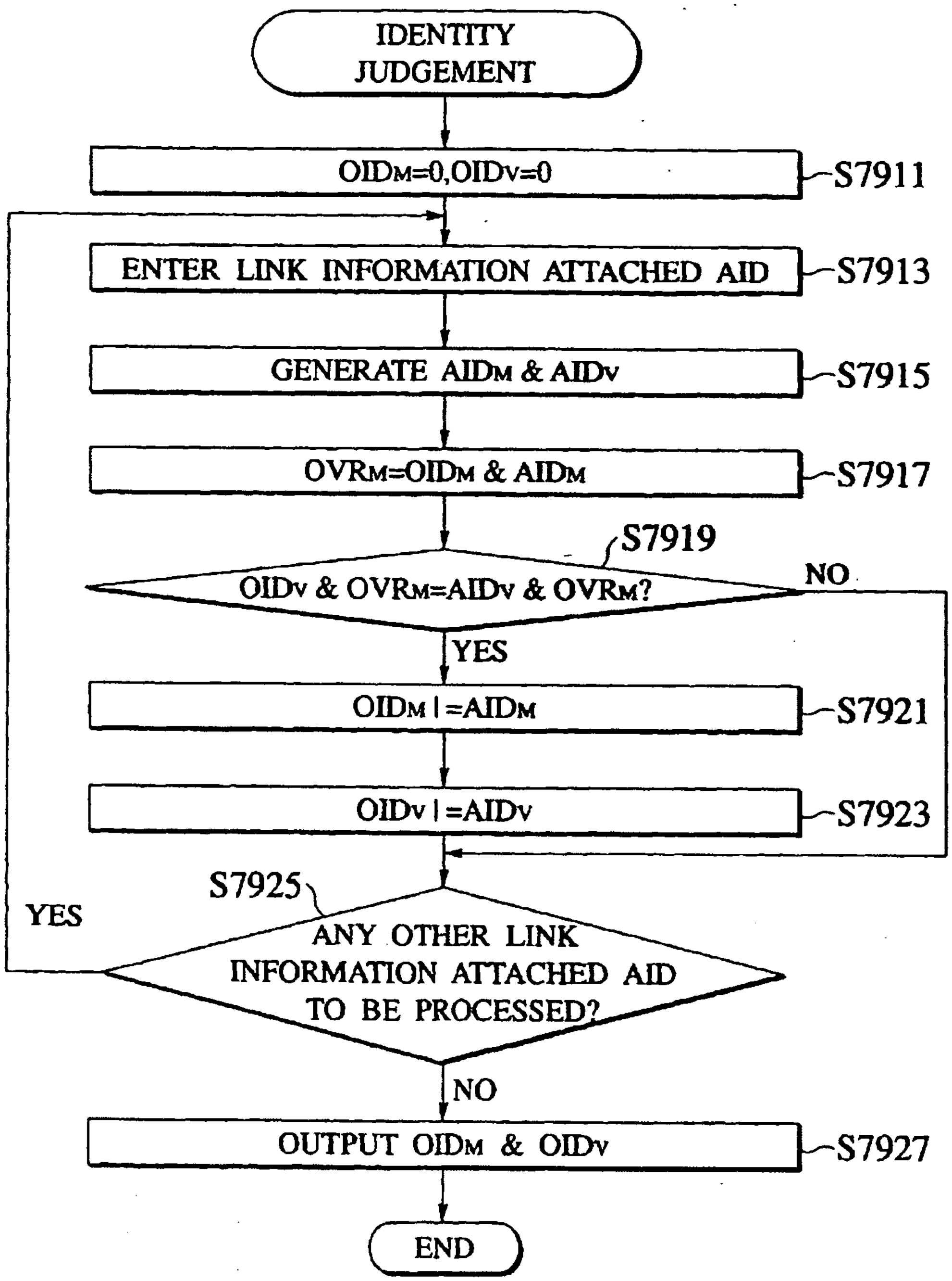

IDENTITY JUDGEMENT
OIDM=0,OIDV=0
S7911
ENTER LINK INFORMATION ATTACHED AID
S7913
GENERATE AIDM & AIDV
S7915
OVRM=OIDM & AIDM
S7917
S7919
OIDV & OVRM=AIDV & OVRM?
NO
YES
OIDM | =AIDM
S7921
OIDV | =AIDV
S7923
S7925
YES
ANY OTHER LINK INFORMATION ATTACHED AID TO BE PROCESSED?
NO
OUTPUT OIDM & OIDV
S7927
END

FIG.39

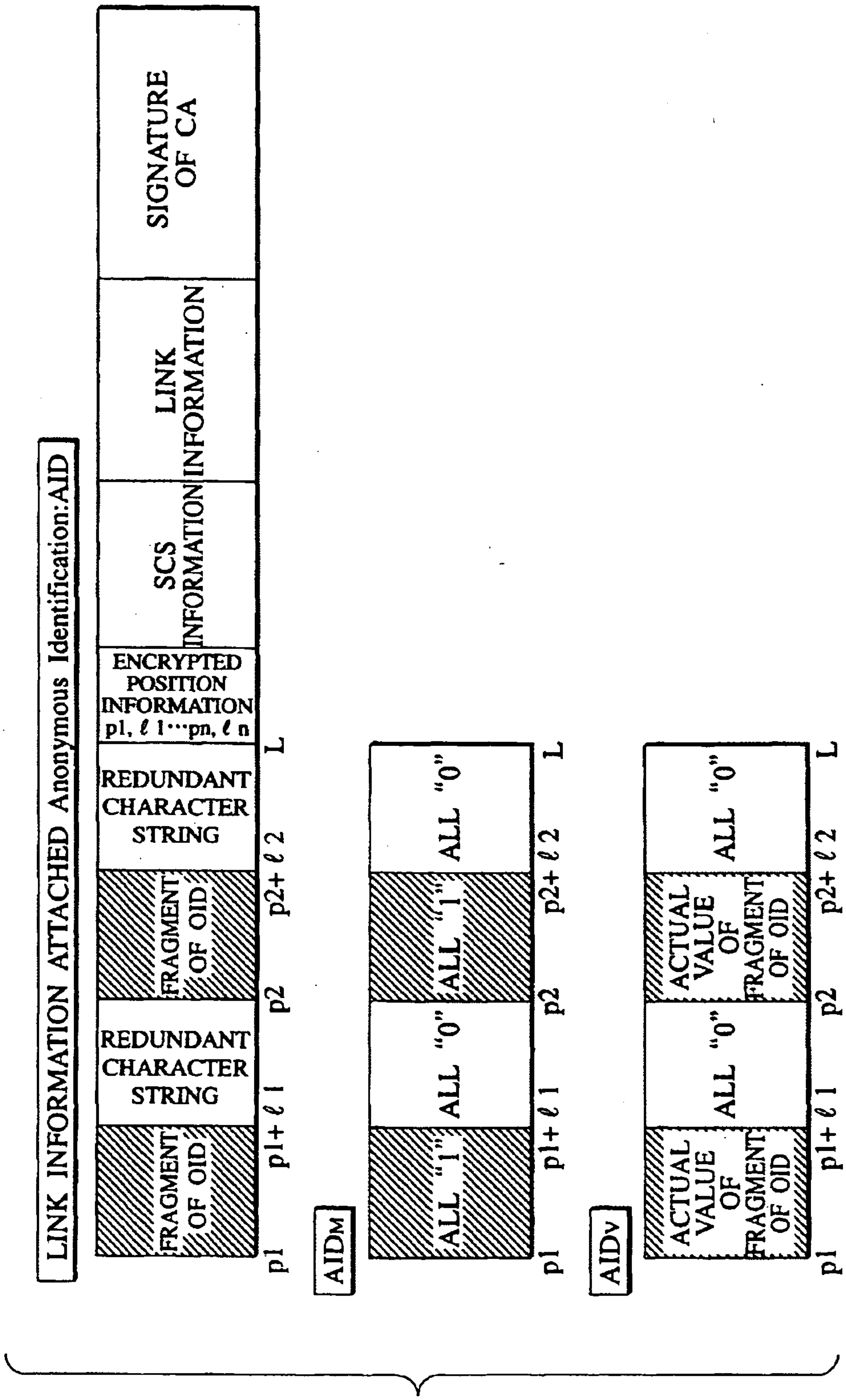

LINK INFORMATION ATTACHED Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
ENCRYPTED POSITION INFORMATION p1, ℓ 1···pn, ℓ n
SCS INFORMATION
LINK INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
p2
p2+ ℓ 2
L
AIDM
ALL "1"
ALL "0"
ALL "1"
ALL "0"
p1
p1+ ℓ 1
p2
p2+ ℓ 2
L
AIDV
ACTUAL VALUE OF FRAGMENT OF OID
ALL "0"
ACTUAL VALUE OF FRAGMENT OF OID
ALL "0"
p1
p1+ ℓ 1
p2
p2+ ℓ 2
L

FIG.40

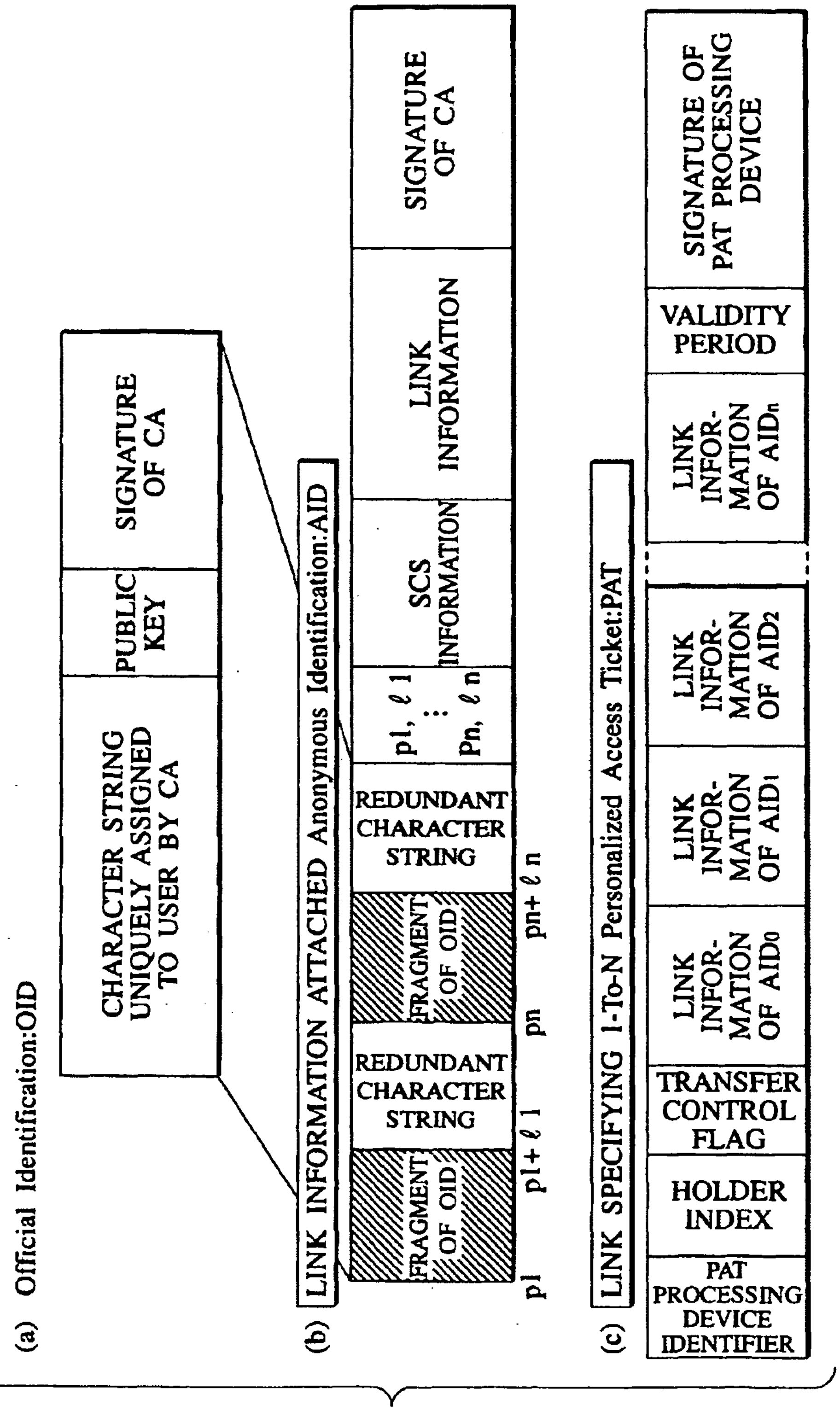

(a) Official Identification:OID
CHARACTER STRING UNIQUELY ASSIGNED TO USER BY CA
PUBLIC KEY
SIGNATURE OF CA
(b) LINK INFORMATION ATTACHED Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
pl, ℓ 1
:
Pn, ℓ n
SCS INFORMATION
LINK INFORMATION
SIGNATURE OF CA
pl
pl+ ℓ 1
pn
pn+ ℓ n
(c) LINK SPECIFYING 1-To-N Personalized Access Ticket:PAT
PAT PROCESSING DEVICE IDENTIFIER
HOLDER INDEX
TRANSFER CONTROL FLAG
LINK INFOR-MATION OF AID0
LINK INFOR-MATION OF AID1
LINK INFOR-MATION OF AID2
LINK INFOR-MATION OF AIDn
VALIDITY PERIOD
SIGNATURE OF PAT PROCESSING DEVICE

FIG.41

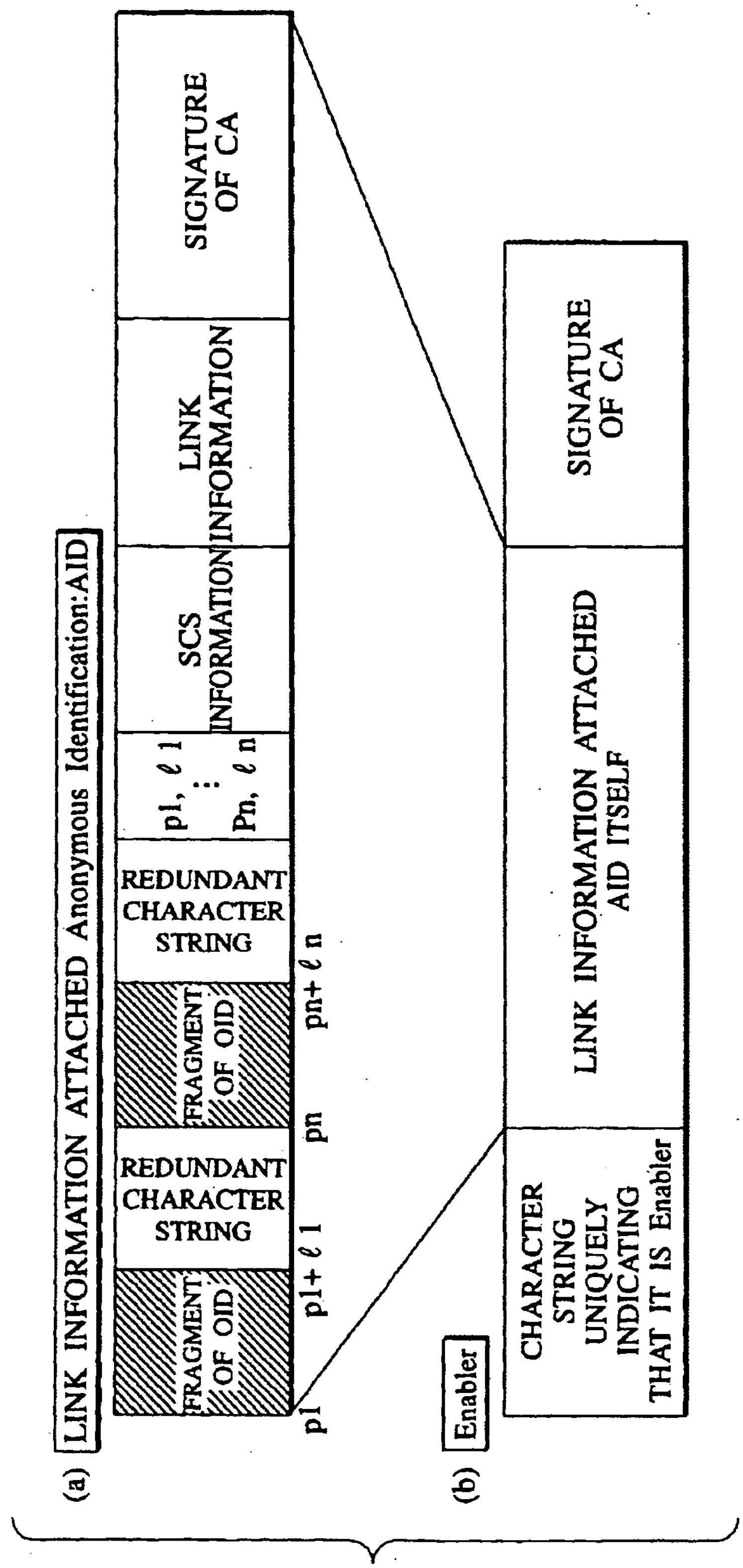
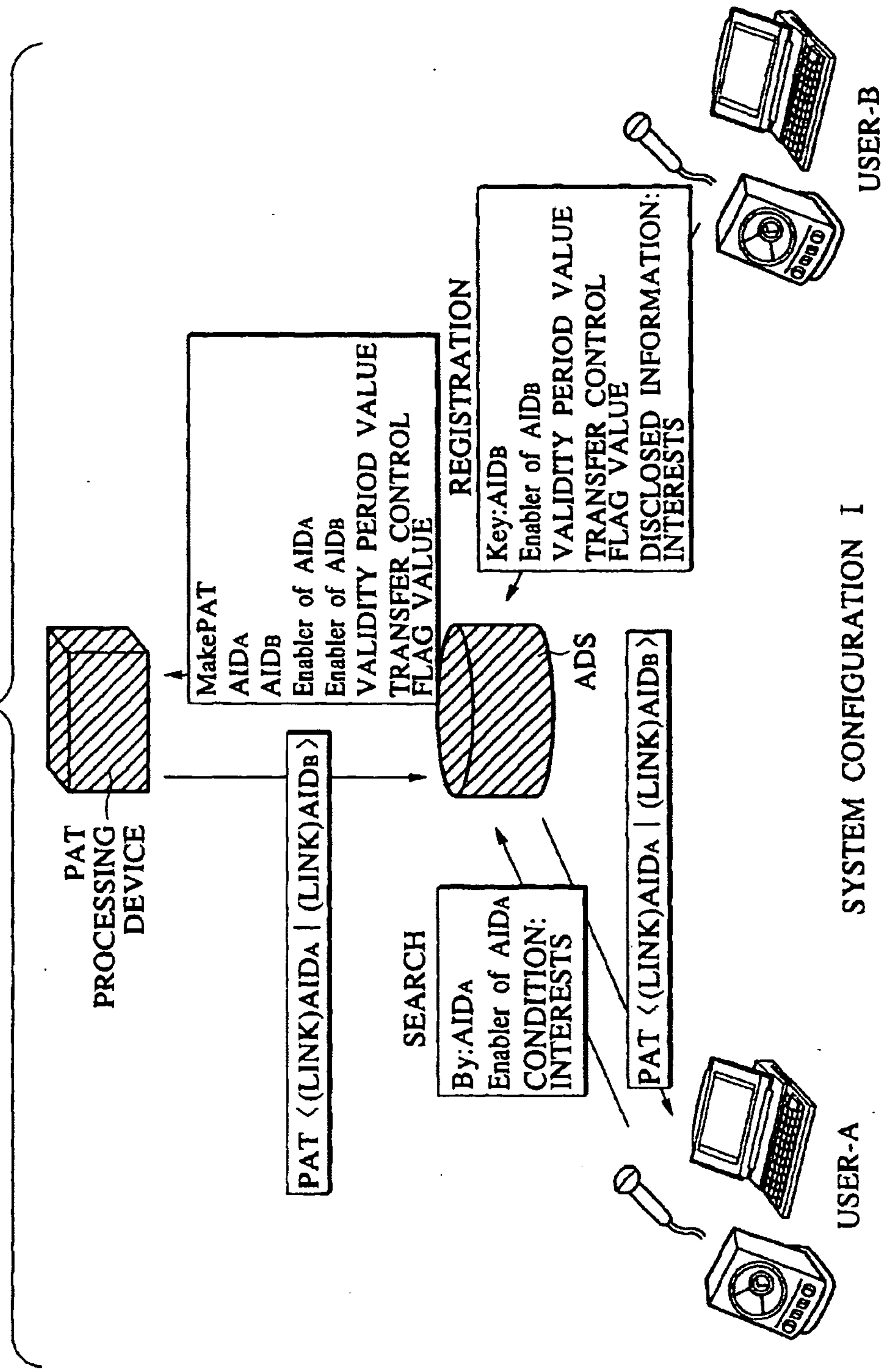
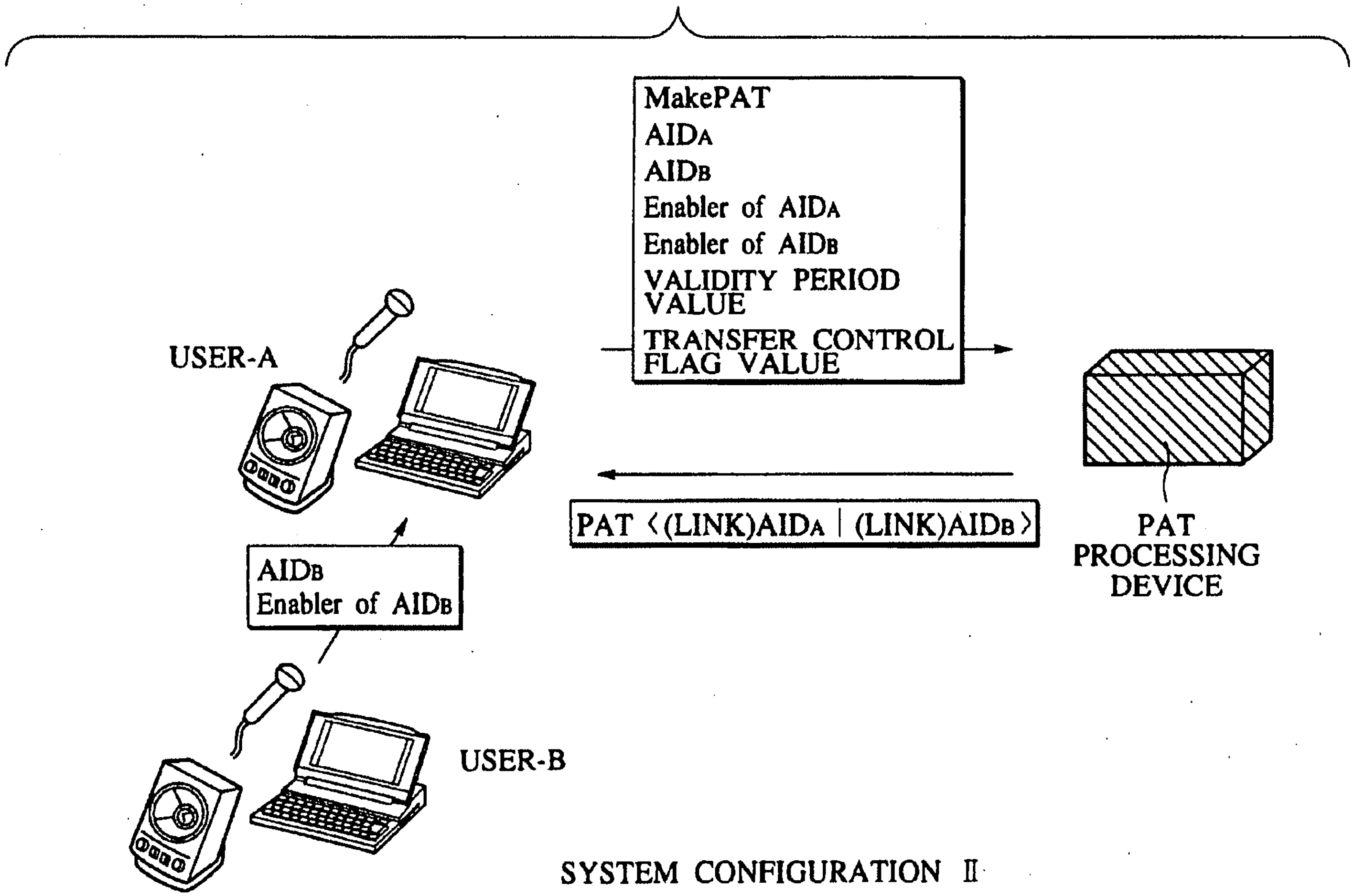

(a)
LINK INFORMATION ATTACHED Anonymous Identification:AID
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
FRAGMENT OF OID
REDUNDANT CHARACTER STRING
p1, ℓ 1
:
Pn, ℓ n
SCS INFORMATION
LINK INFORMATION
SIGNATURE OF CA
p1
p1+ ℓ 1
pn
pn+ ℓ n
(b)
Enabler
CHARACTER STRING UNIQUELY INDICATING THAT IT IS Enabler
LINK INFORMATION ATTACHED AID ITSELF
SIGNATURE OF CA

FIG.42



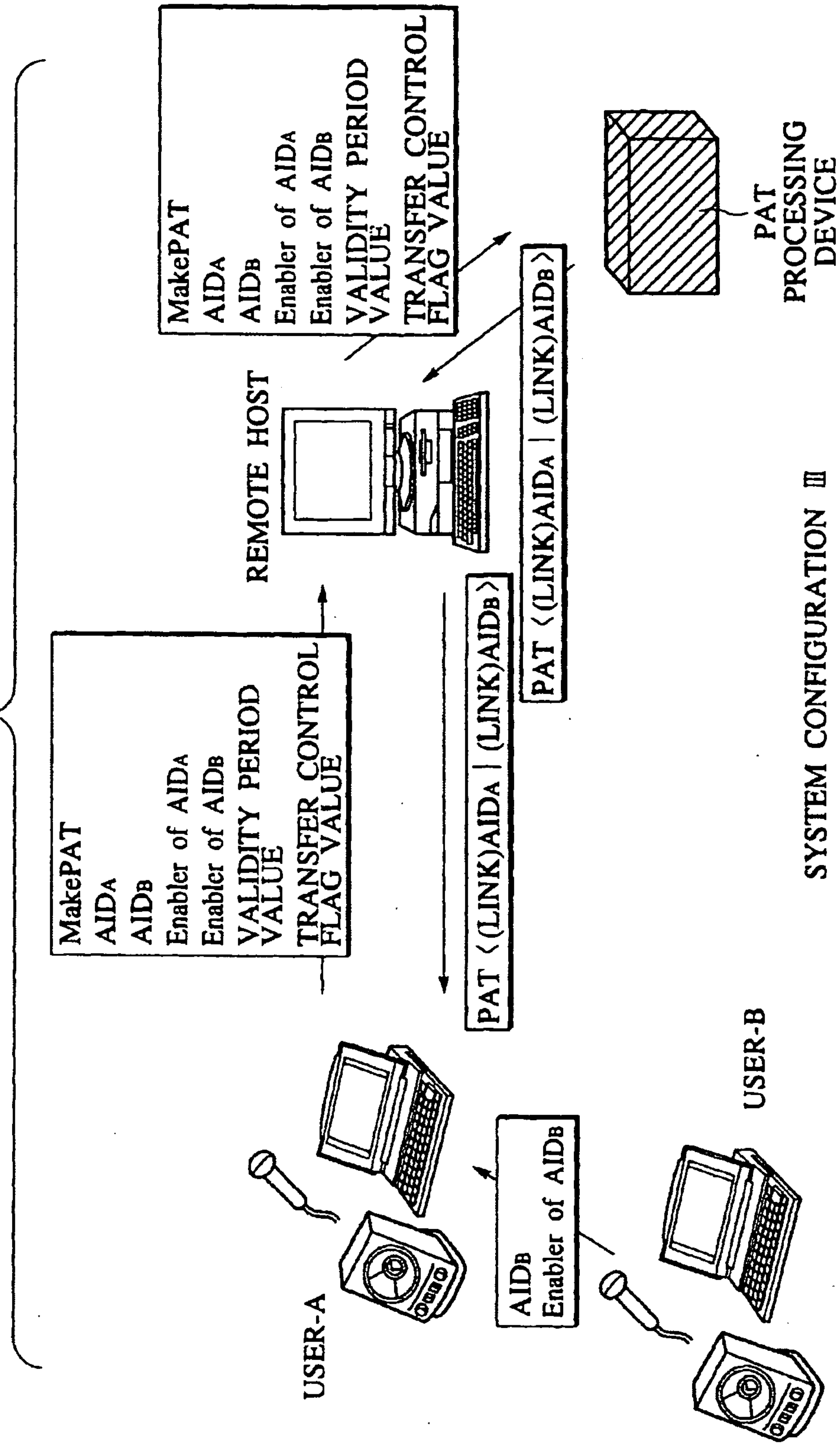
SYSTEM CONFIGURATION I

FIG.43



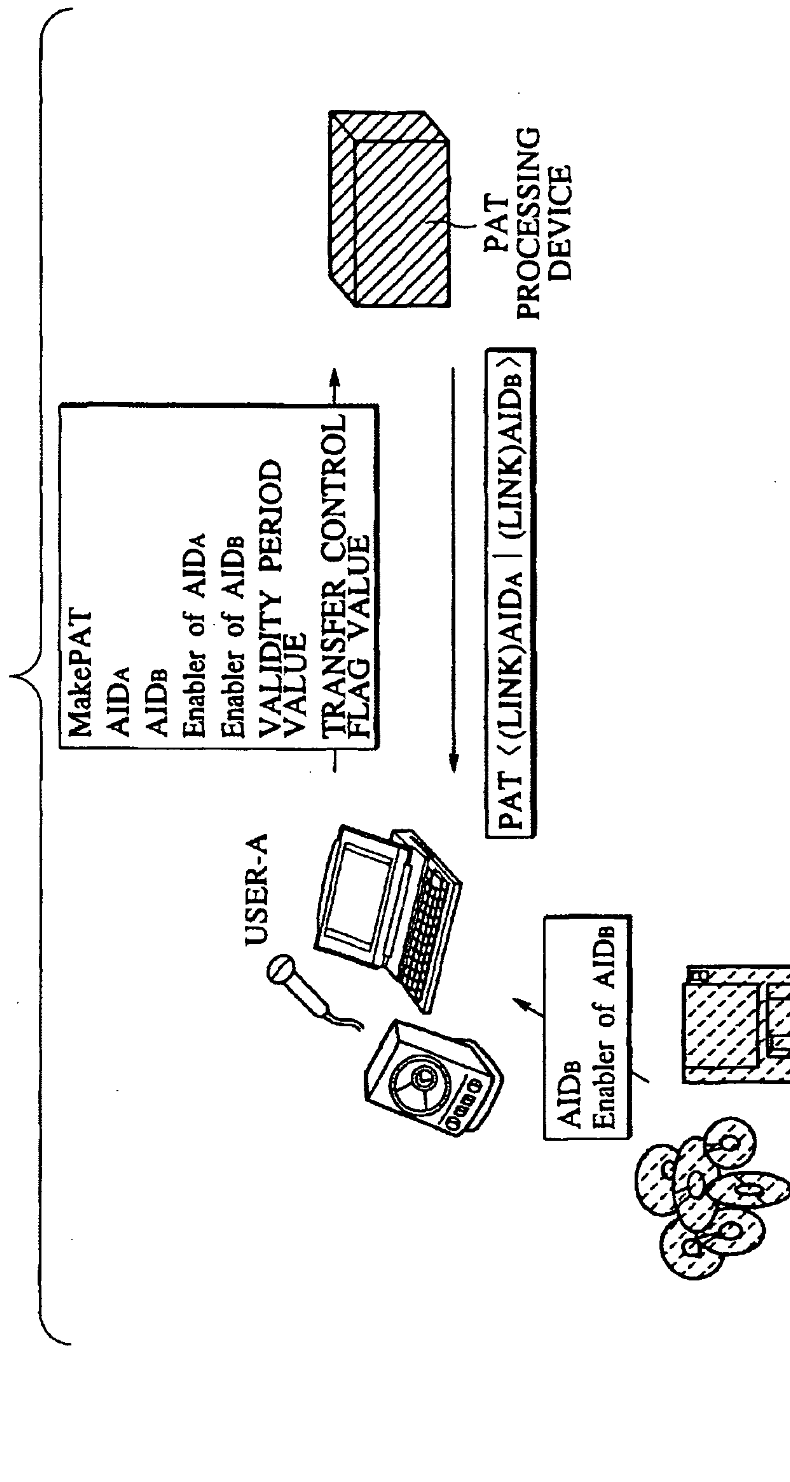
SYSTEM CONFIGURATION II

FIG.44



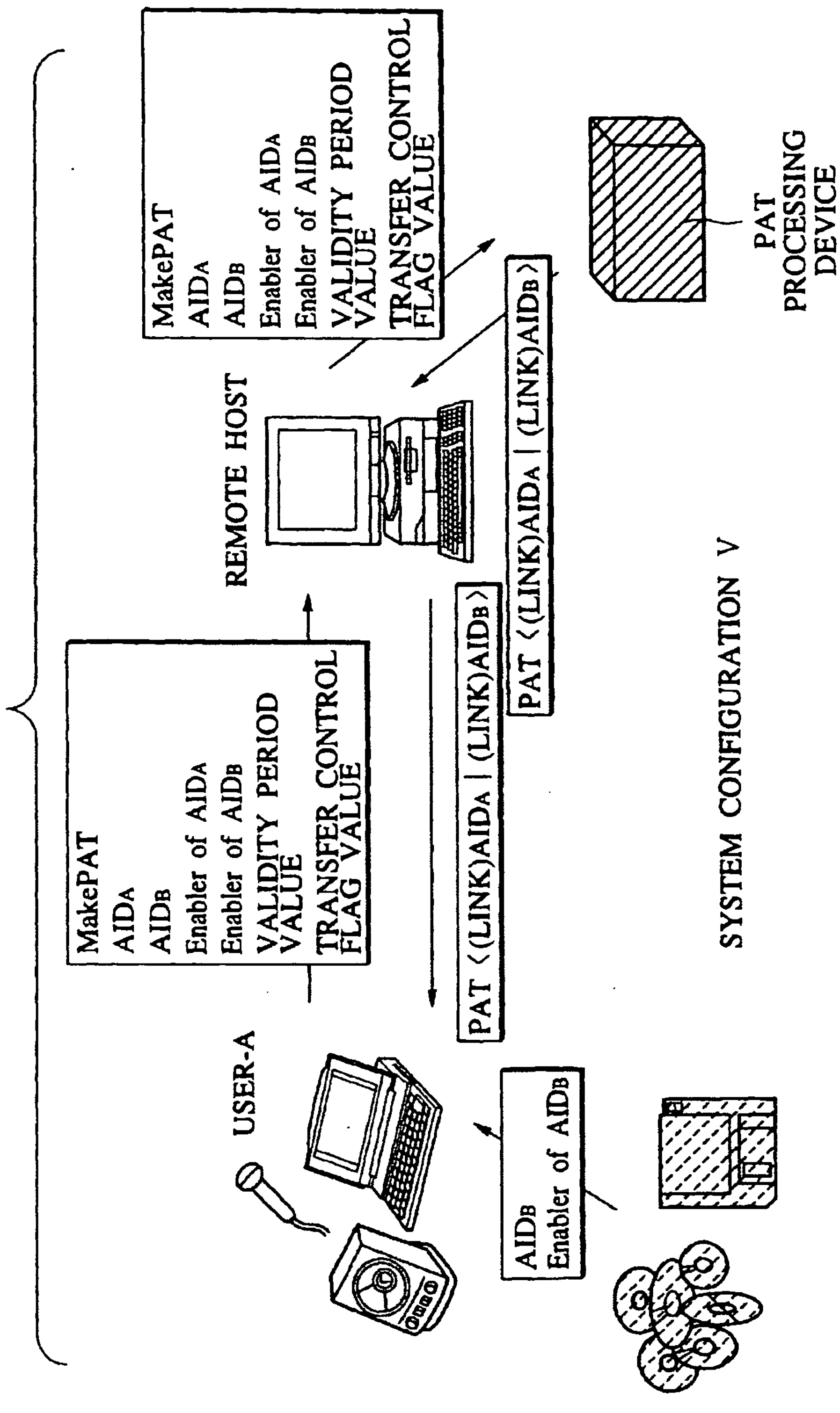
SYSTEM CONFIGURATION Ⅲ

FIG.45



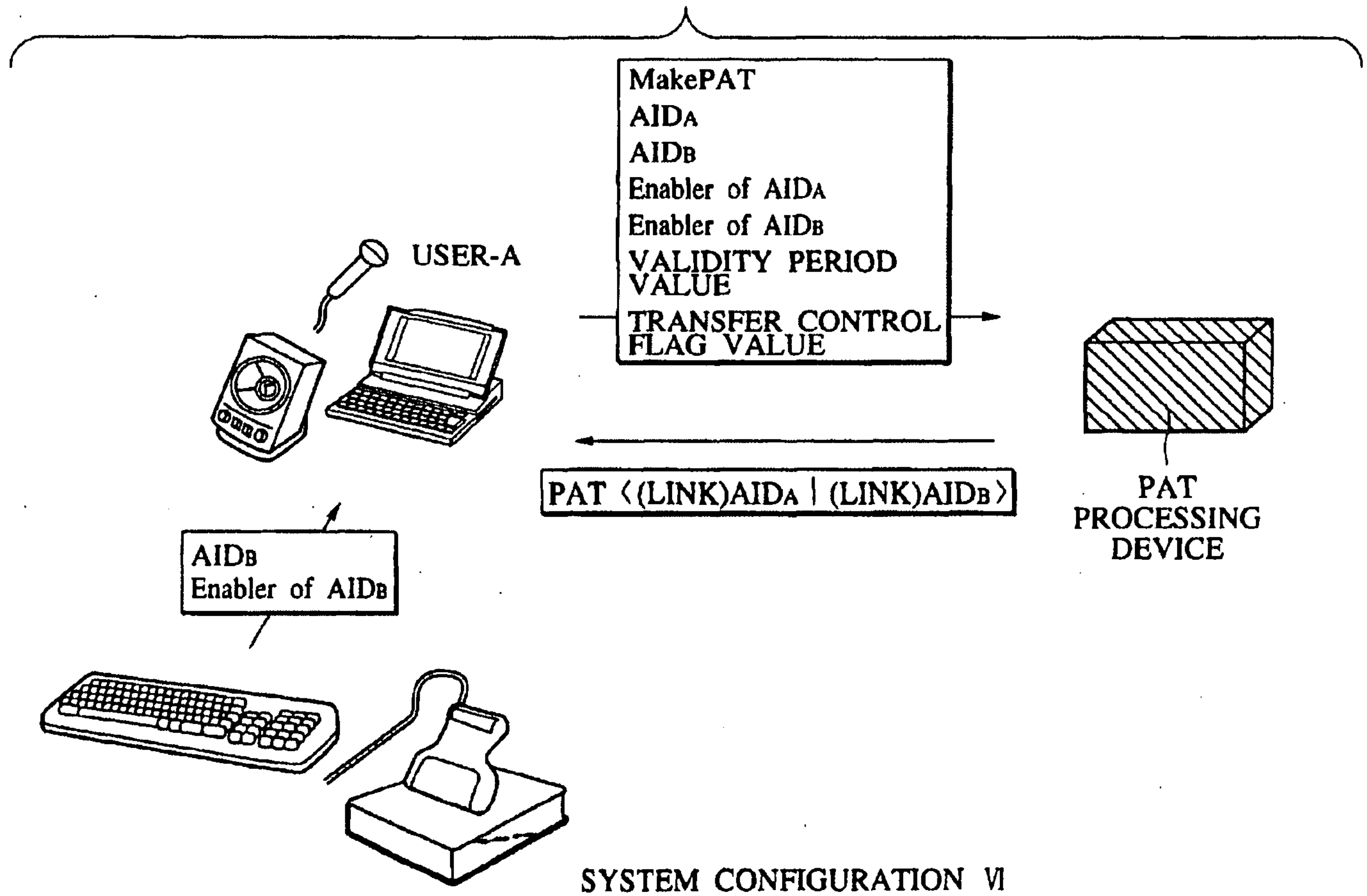
SYSTEM CONFIGURATION IV

FIG.46



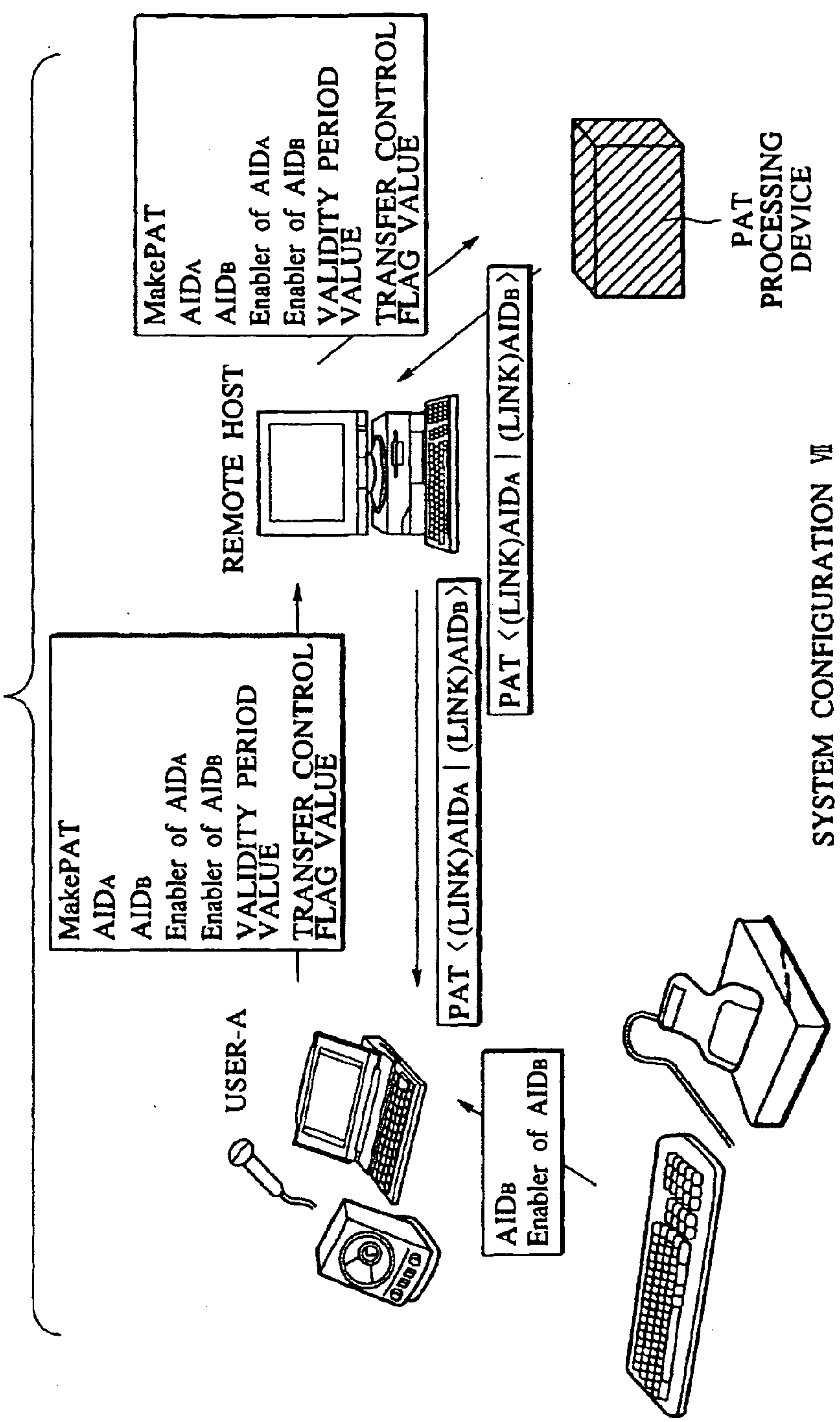
SYSTEM CONFIGURATION V

FIG.47



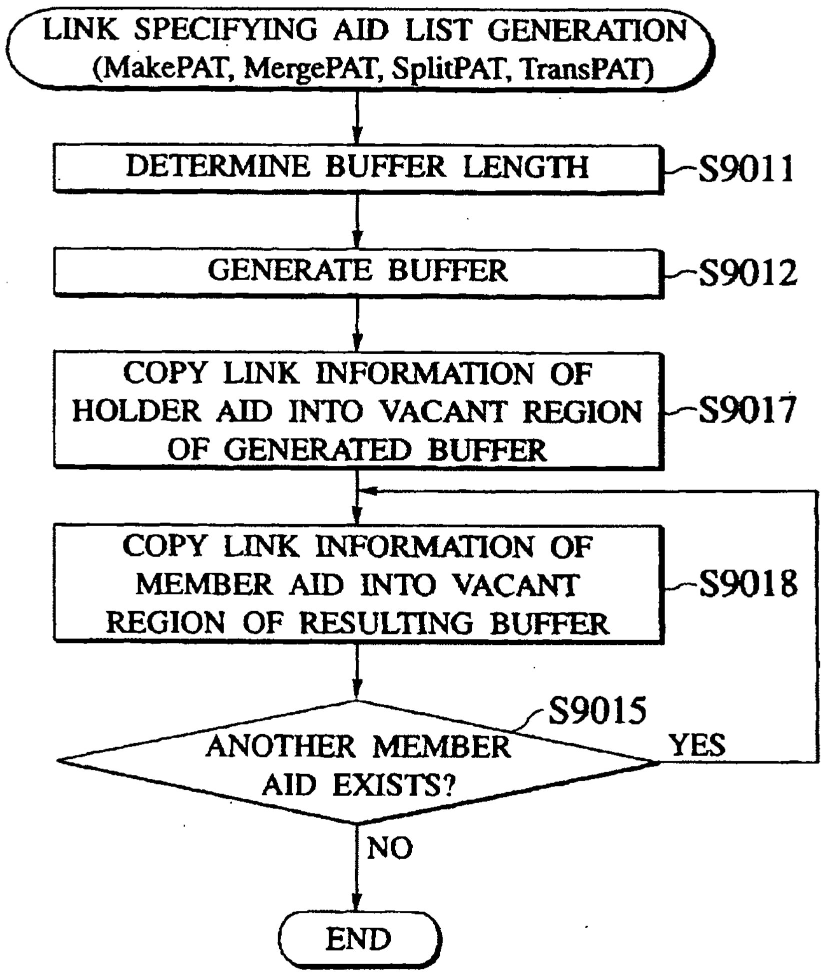
SYSTEM CONFIGURATION VI

FIG.48



SYSTEM CONFIGURATION VII

FIG.49


LINK SPECIFYING AID LIST GENERATION
(MakePAT, MergePAT, SplitPAT, TransPAT)
DETERMINE BUFFER LENGTH
S9011
GENERATE BUFFER
S9012
COPY LINK INFORMATION OF HOLDER AID INTO VACANT REGION OF GENERATED BUFFER
S9017
COPY LINK INFORMATION OF MEMBER AID INTO VACANT REGION OF RESULTING BUFFER
S9018
S9015
ANOTHER MEMBER AID EXISTS?
YES
NO
END

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 964 572 A1

(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

(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

(71) Applicant:
CANAL+ Société Anonyme
75711 Paris Cedex 15 (FR)

(72) Inventor: Declerck, Christophe
28210 Senantes (FR)

(74) Representative:
Cozens, Paul Dennis et al
Mathys & Squire
100 Grays Inn Road
London WC1X 8AL (GB)

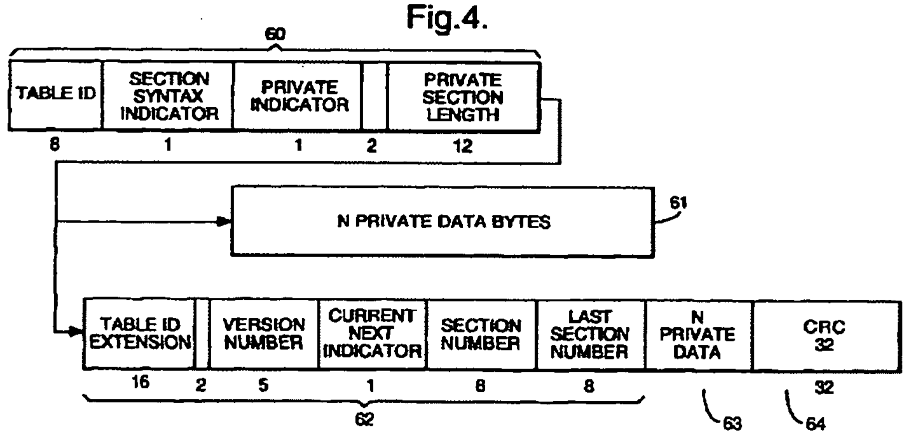
(54) Decoder and security module for a digital transmission system

(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.

The invention equally extends to a portable security module 30 including a memory holding such data as is 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.

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.

[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.

[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.

[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.

[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.

[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.

[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).

[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.

[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.

[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.

[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).

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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:

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 modemmed 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 modemmed 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.

[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.

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.

[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).

[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.

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.

[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.

[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.

[0061] The EMMs are passed to the Ciphering Unit (CU) 24 for ciphering 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.

[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.

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.

[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.

<u>Private Section Data</u>

[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 accomodated 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 ouput 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 ± 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 occured.
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 occured. 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 bib 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 <u>not</u> matching target.

Examples of use:

Example 1:

[0100]

Filter_all_sections(5, 0x8C7C453AA8BBFF00, 0XFF557FFFEEFFFF00, 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]

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.

[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:

Table id	8 bits
Zero	11 bits
Filter id	5 bits
CA specific header field	56 bits
CA message	N*8 bits

[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:

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

Number_of_filters describes the number of filters to be set in this instruction.
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.
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.

[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

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.

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.

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).

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.

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.

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.

10. A decoder as claimed in any preceding claim adapted to process encrypt and/or decrypt communications to and from the portable security module.

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.

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.

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.

Fig.1.

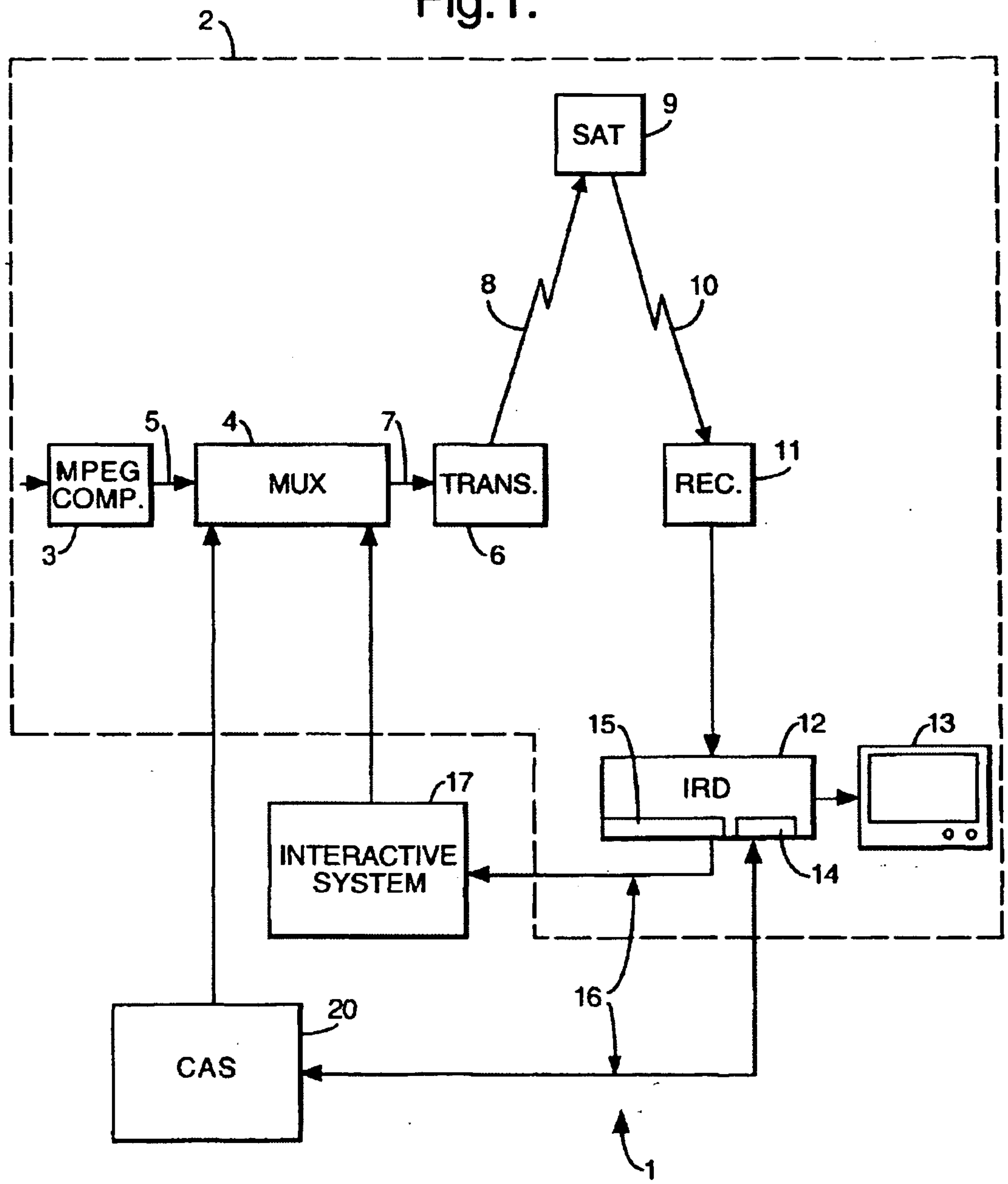

2
SAT
9
8
10
5
4
7
MPEG
COMP.
MUX
TRANS.
REC.
11
3
6
15
12
13
IRD
17
INTERACTIVE
SYSTEM
14
16
20
CAS
1

Fig.2.

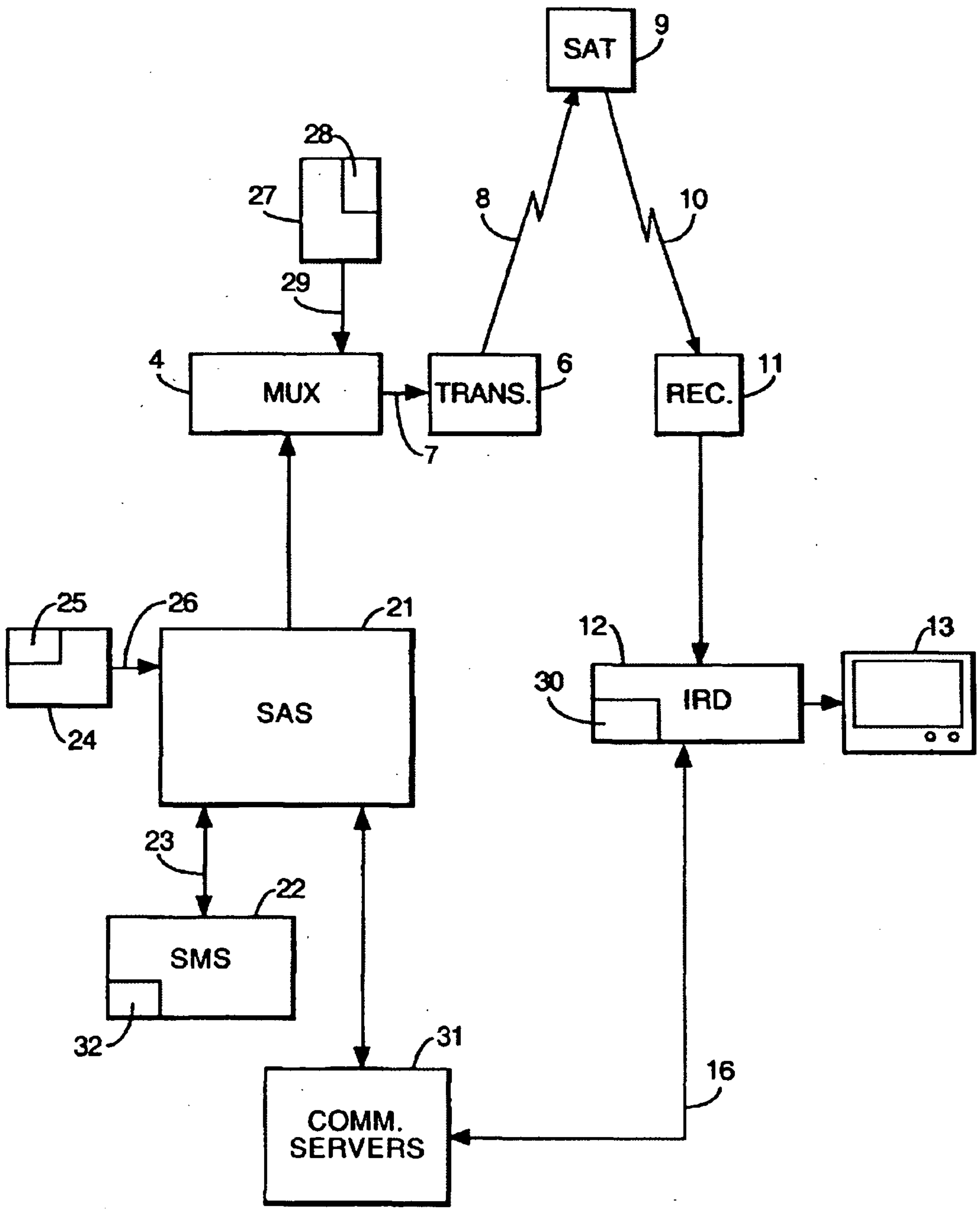

SAT
9
28
27
8
10
29
4
MUX
TRANS.
6
REC.
11
7
25
26
21
12
13
30
IRD
SAS
24
23
22
SMS
32
31
16
COMM.
SERVERS

Fig.3.

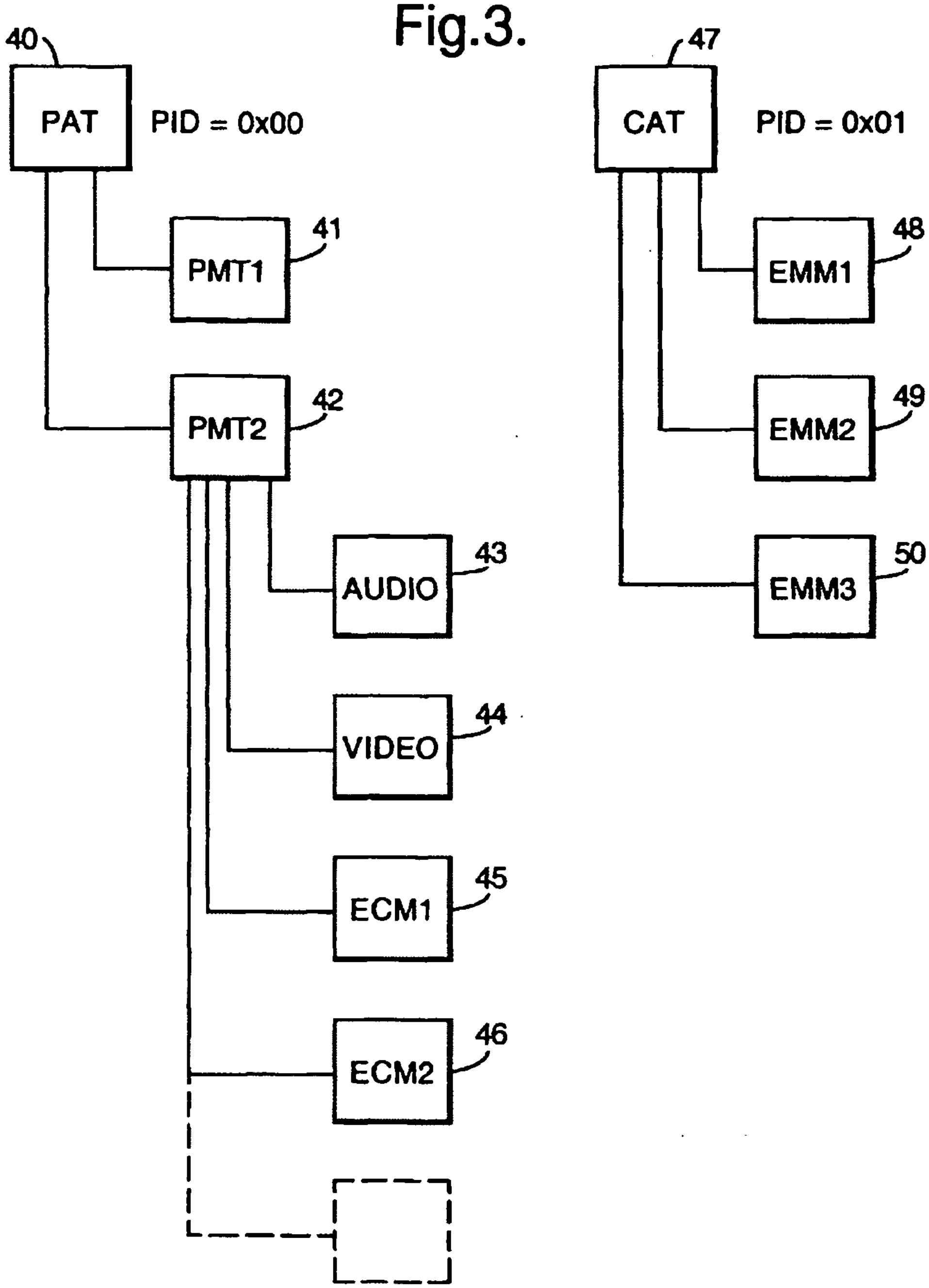

40
PAT
PID = 0x00
PMT1
41
PMT2
42
AUDIO
43
VIDEO
44
ECM1
45
ECM2
46
47
CAT
PID = 0x01
EMM1
48
EMM2
49
EMM3
50

Fig.4.

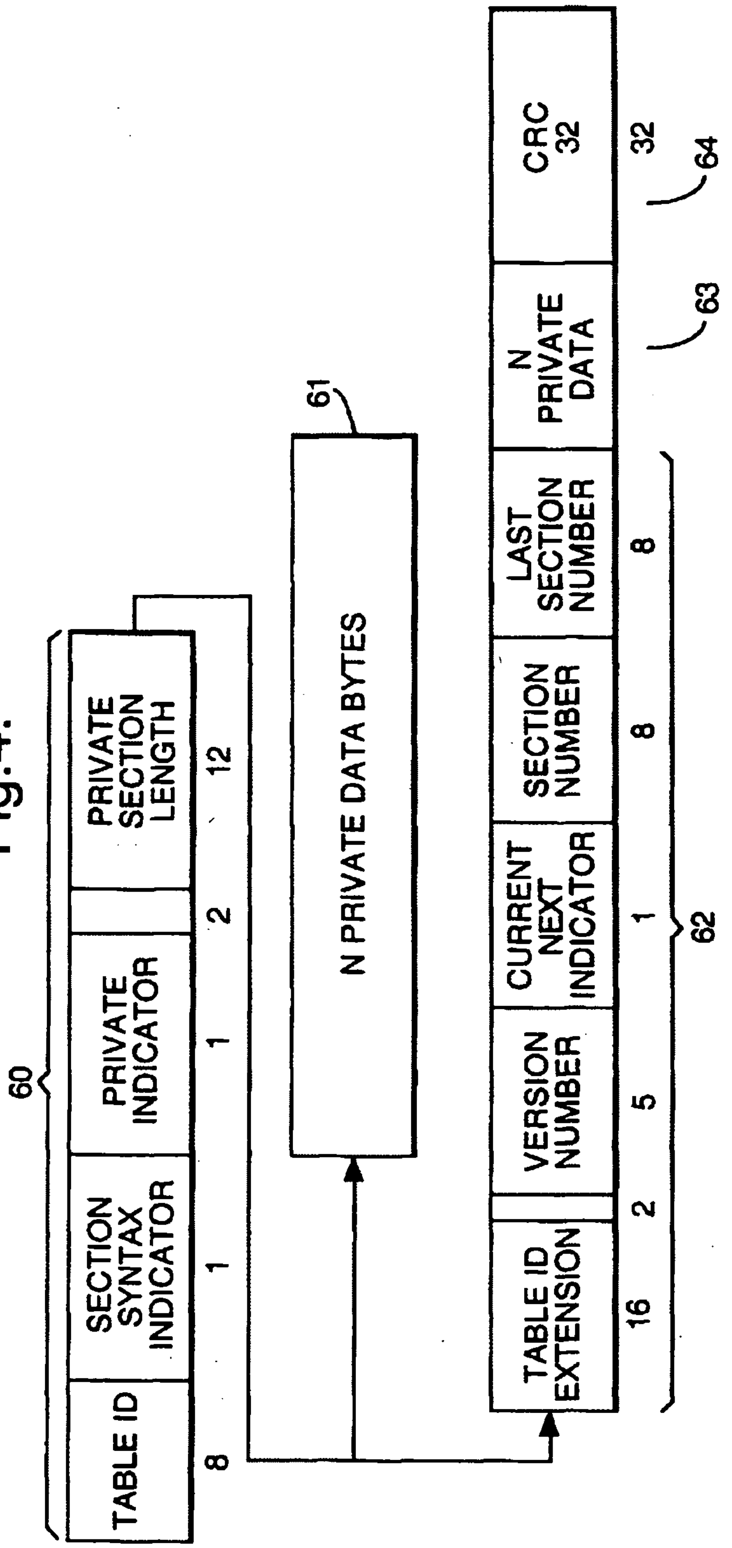

60
TABLE ID
SECTION SYNTAX INDICATOR
PRIVATE INDICATOR
PRIVATE SECTION LENGTH
8
1
1
2
12
N PRIVATE DATA BYTES
61
TABLE ID EXTENSION
VERSION NUMBER
CURRENT NEXT INDICATOR
SECTION NUMBER
LAST SECTION NUMBER
N PRIVATE DATA
CRC 32
16
2
5
1
8
8
32
62
63
64

Fig.5.

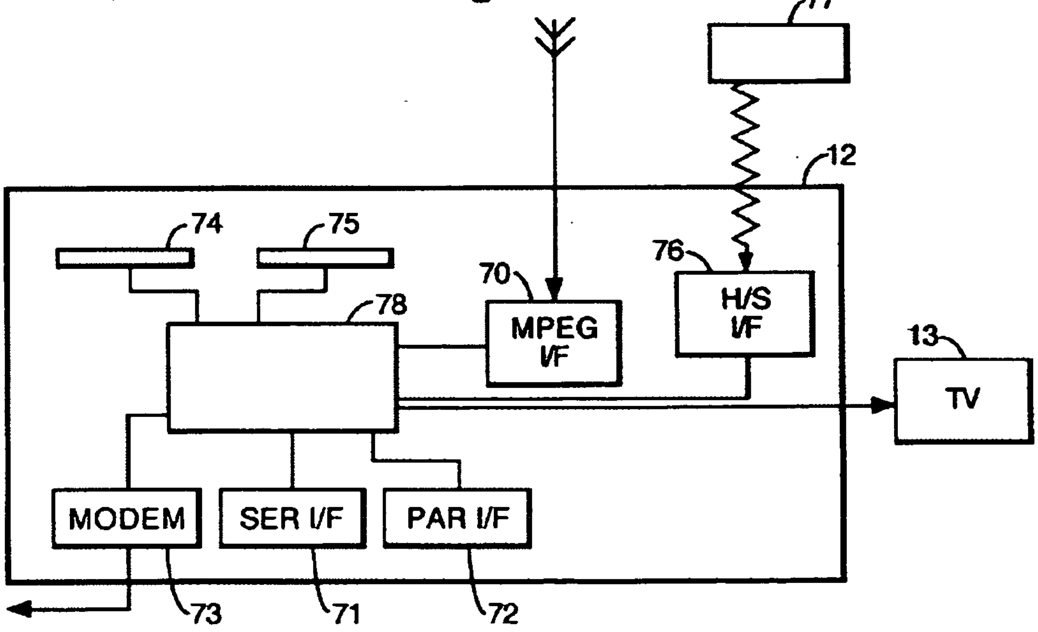


Fig.6.

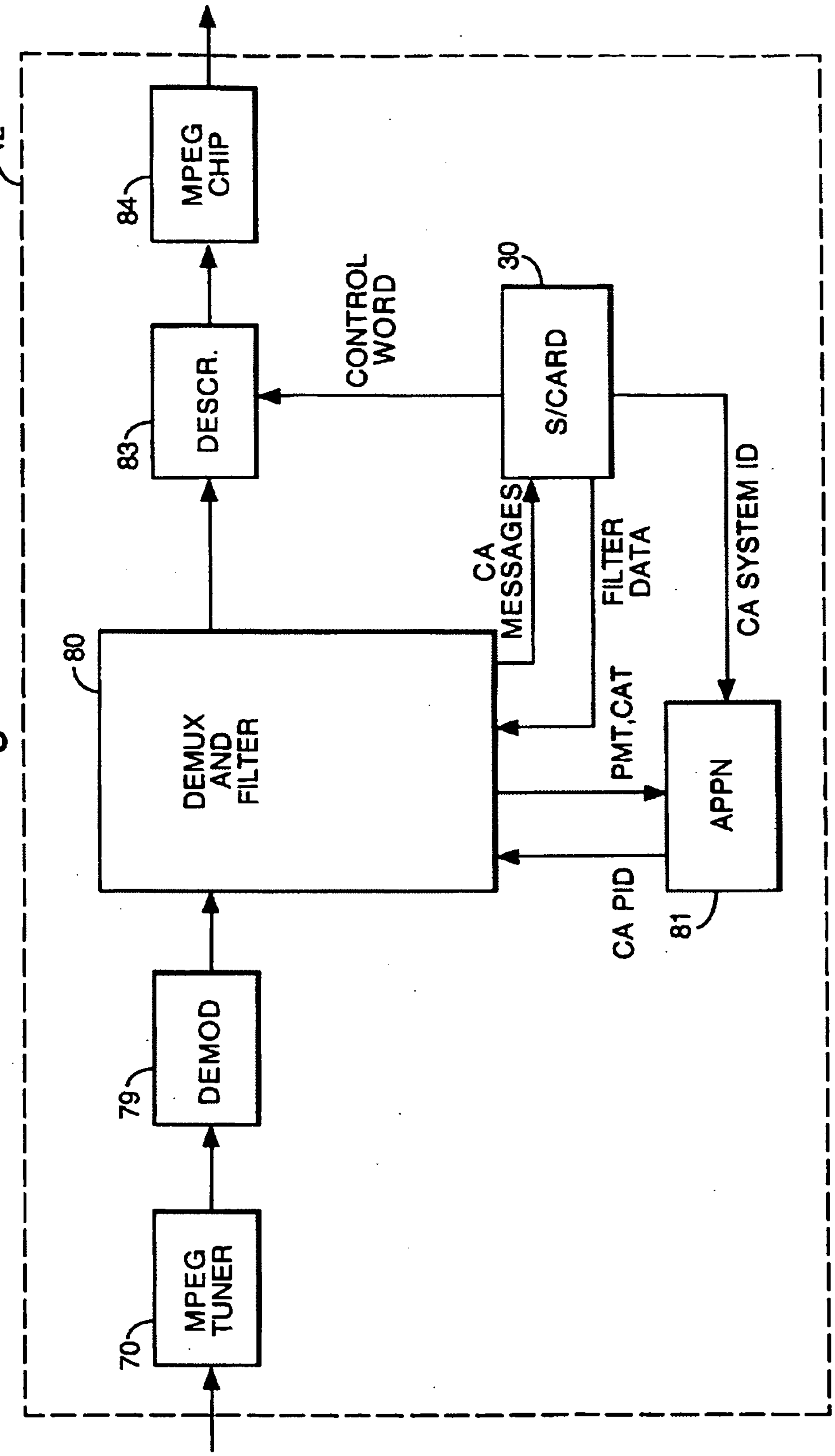
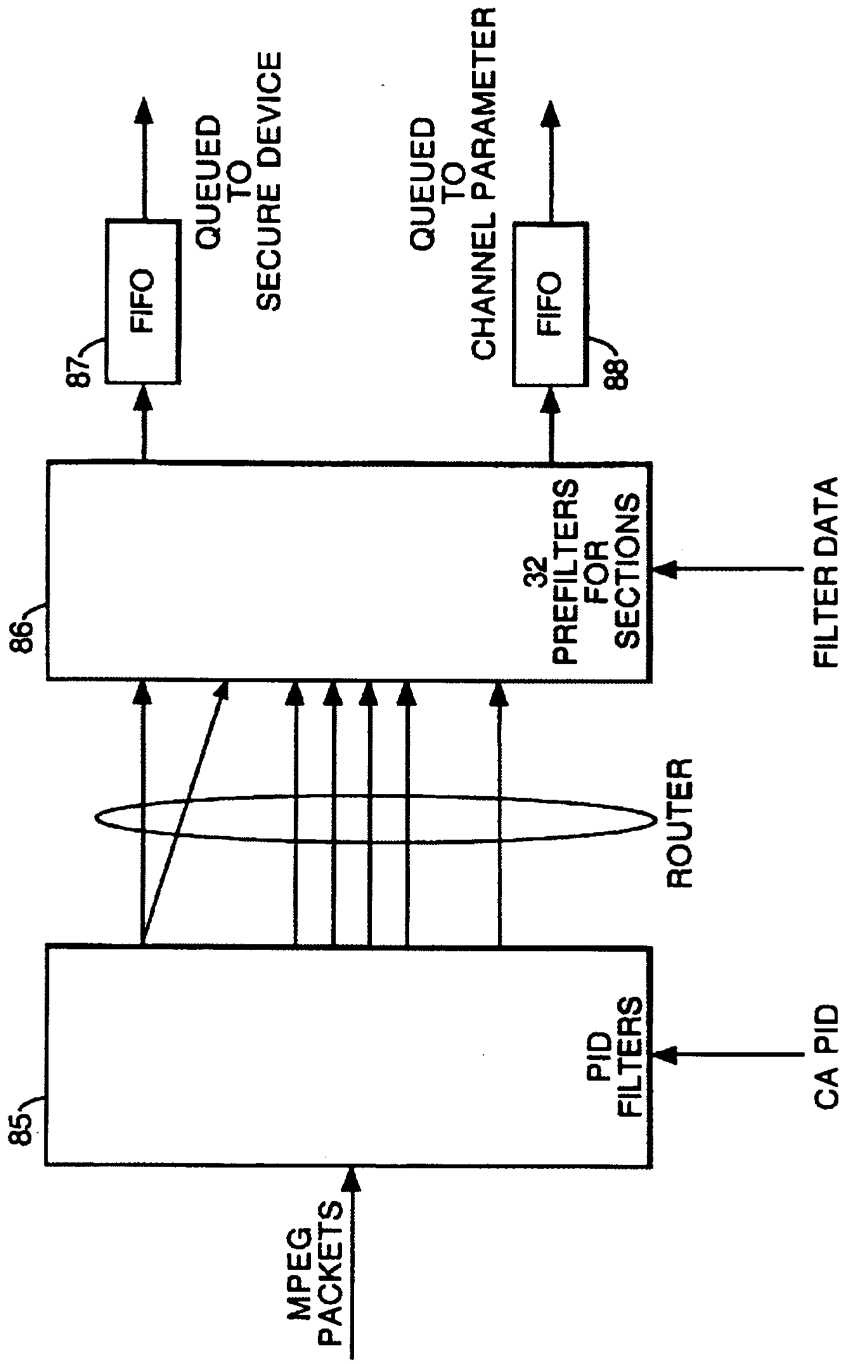

12
MPEG TUNER
70
DEMOD
79
DEMUX AND FILTER
80
DESCR.
83
MPEG CHIP
84
S/CARD
30
APPN
81
CONTROL WORD
CA MESSAGES
FILTER DATA
CA SYSTEM ID
PMT,CAT
CA PID

Fig.7.


MPEG PACKETS
85
PID FILTERS
CA PID
ROUTER
86
32 PREFILTERS FOR SECTIONS
FILTER DATA
87
FIFO
QUEUED TO SECURE DEVICE
88
FIFO
QUEUED TO CHANNEL PARAMETER

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	Date of completion of the search	Examiner
THE HAGUE	3 November 1998	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

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (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

---

(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.

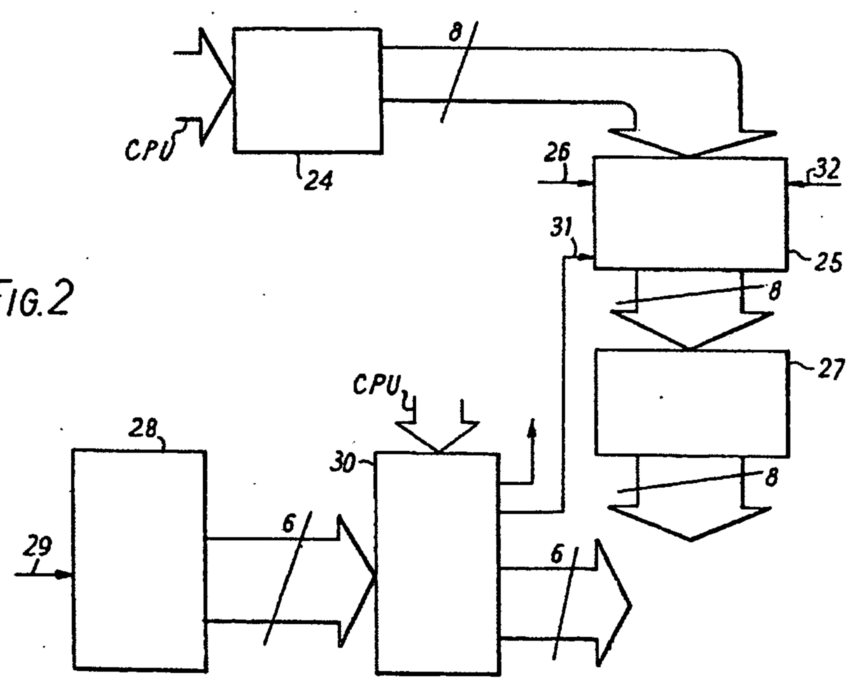


FIG. 2

GB 2 022 969 A

1/2

2022969

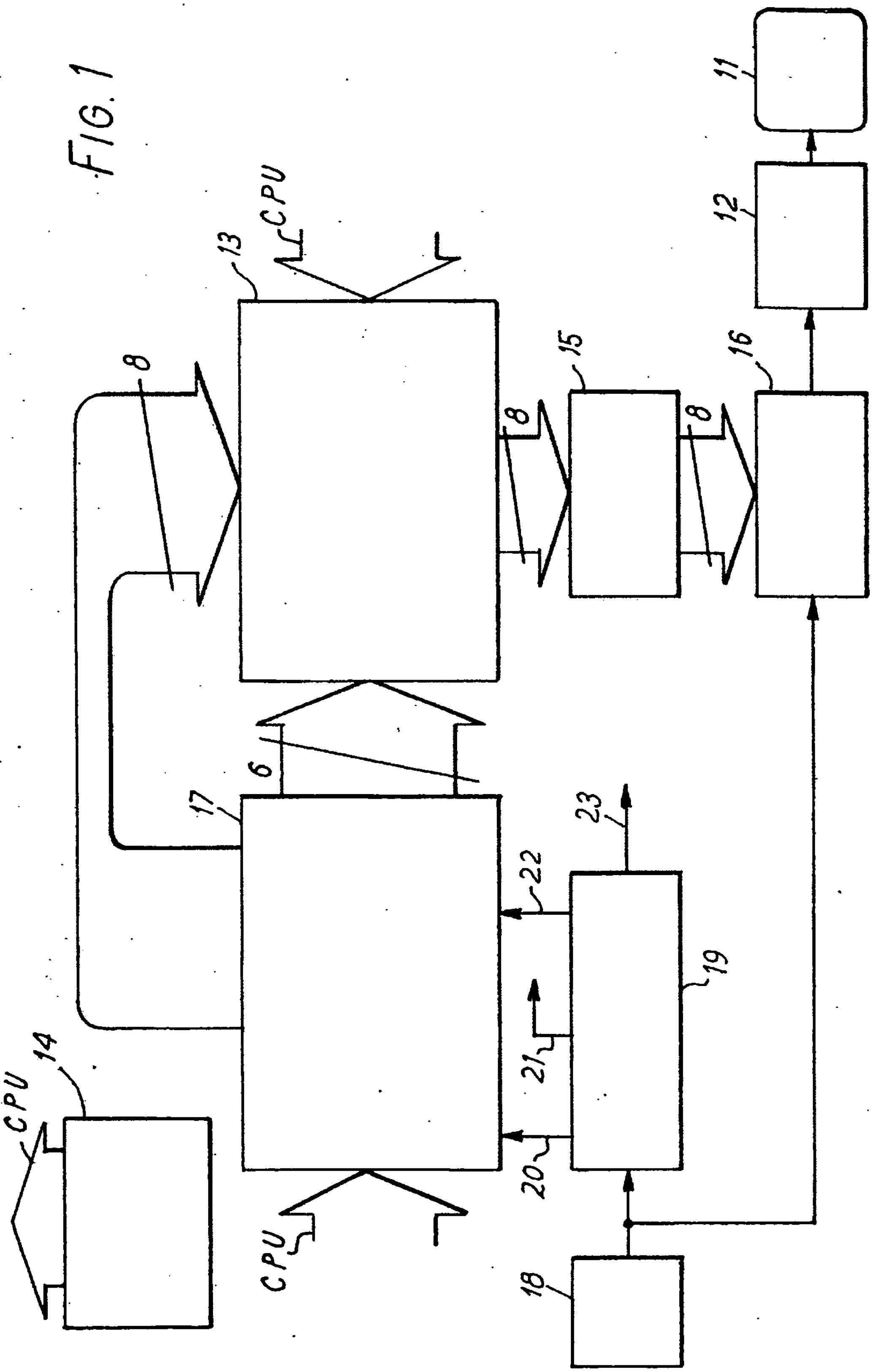


FIG. 1

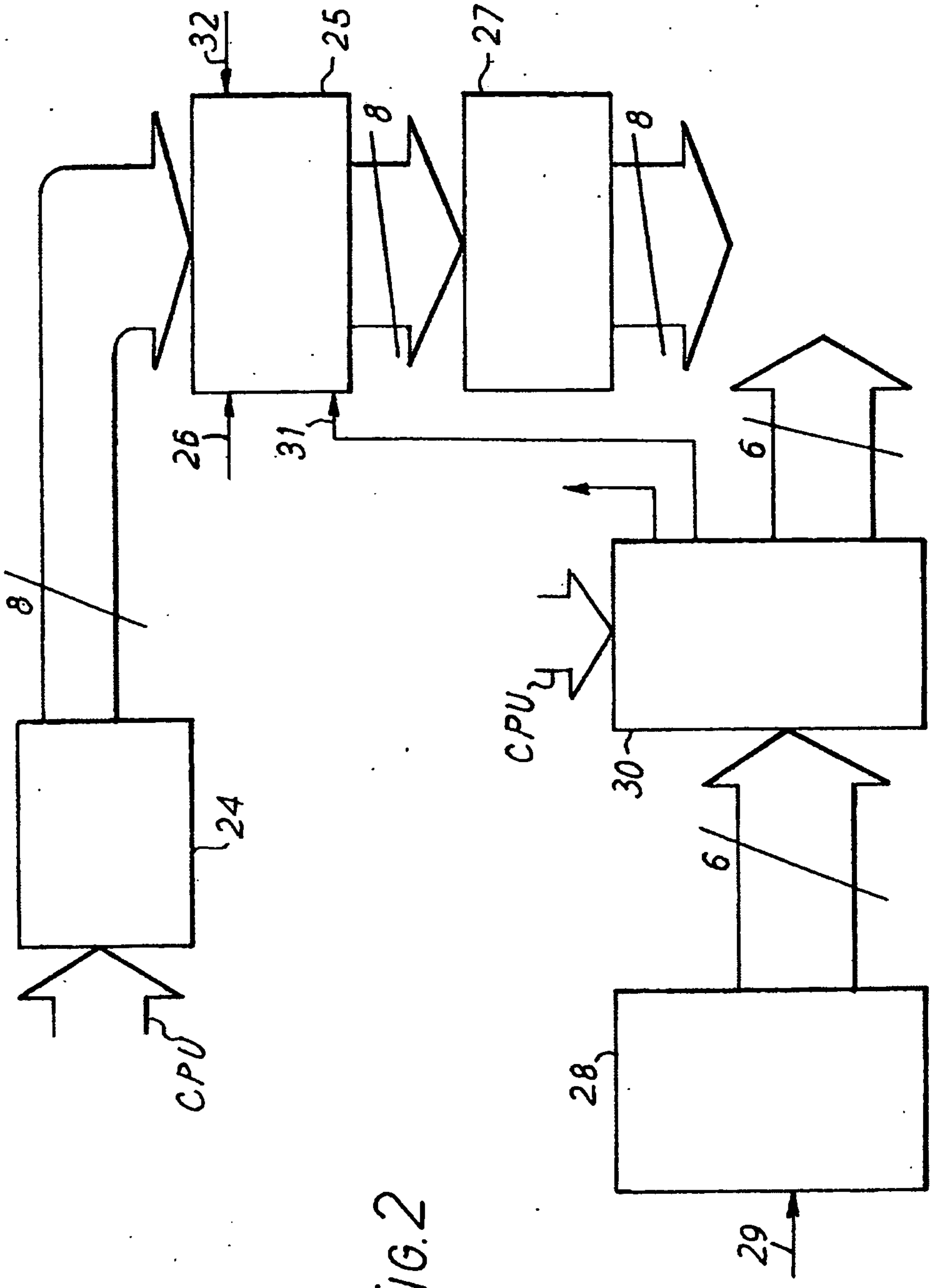


FIG.2

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 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-toserial 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 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.

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.

Printed for Her Majesty's Stationery Office by the Courier Press, Leamington Spa, 1979. Published by the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

(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 CL[7]
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 CL[7] **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.

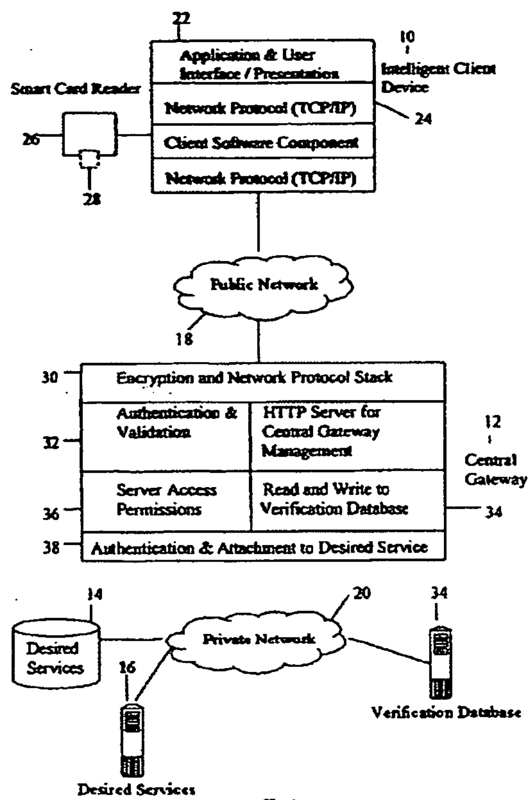


Fig 1

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.

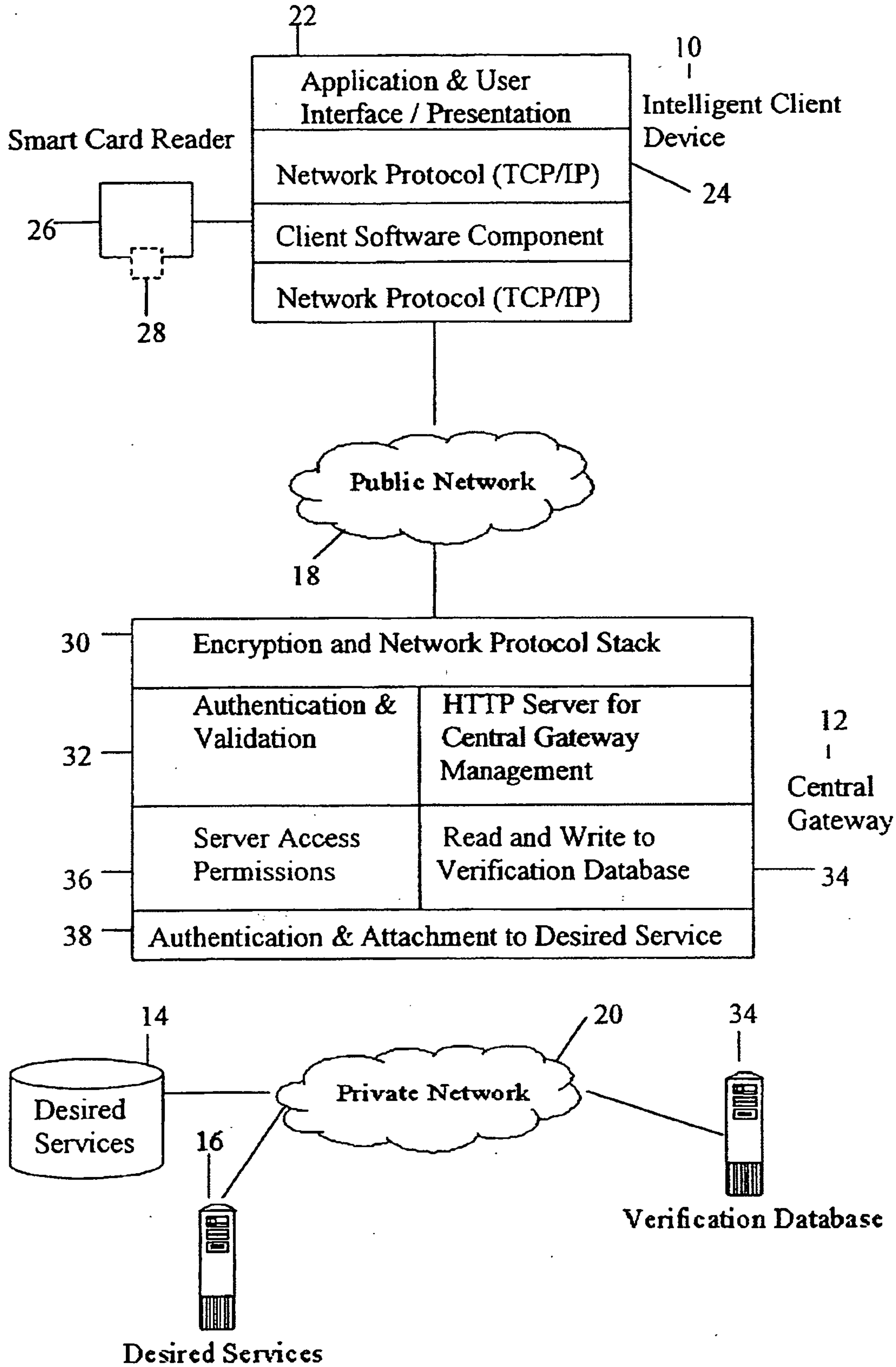

22
10
Application & User Interface / Presentation
Intelligent Client Device
Smart Card Reader
Network Protocol (TCP/IP)
24
26
Client Software Component
Network Protocol (TCP/IP)
28
Public Network
18
30
Encryption and Network Protocol Stack
Authentication & Validation
HTTP Server for Central Gateway Management
12
32
Central Gateway
Server Access Permissions
Read and Write to Verification Database
36
34
38
Authentication & Attachment to Desired Service
14
20
34
Desired Services
Private Network
16
Verification Database
Desired Services

Fig 1

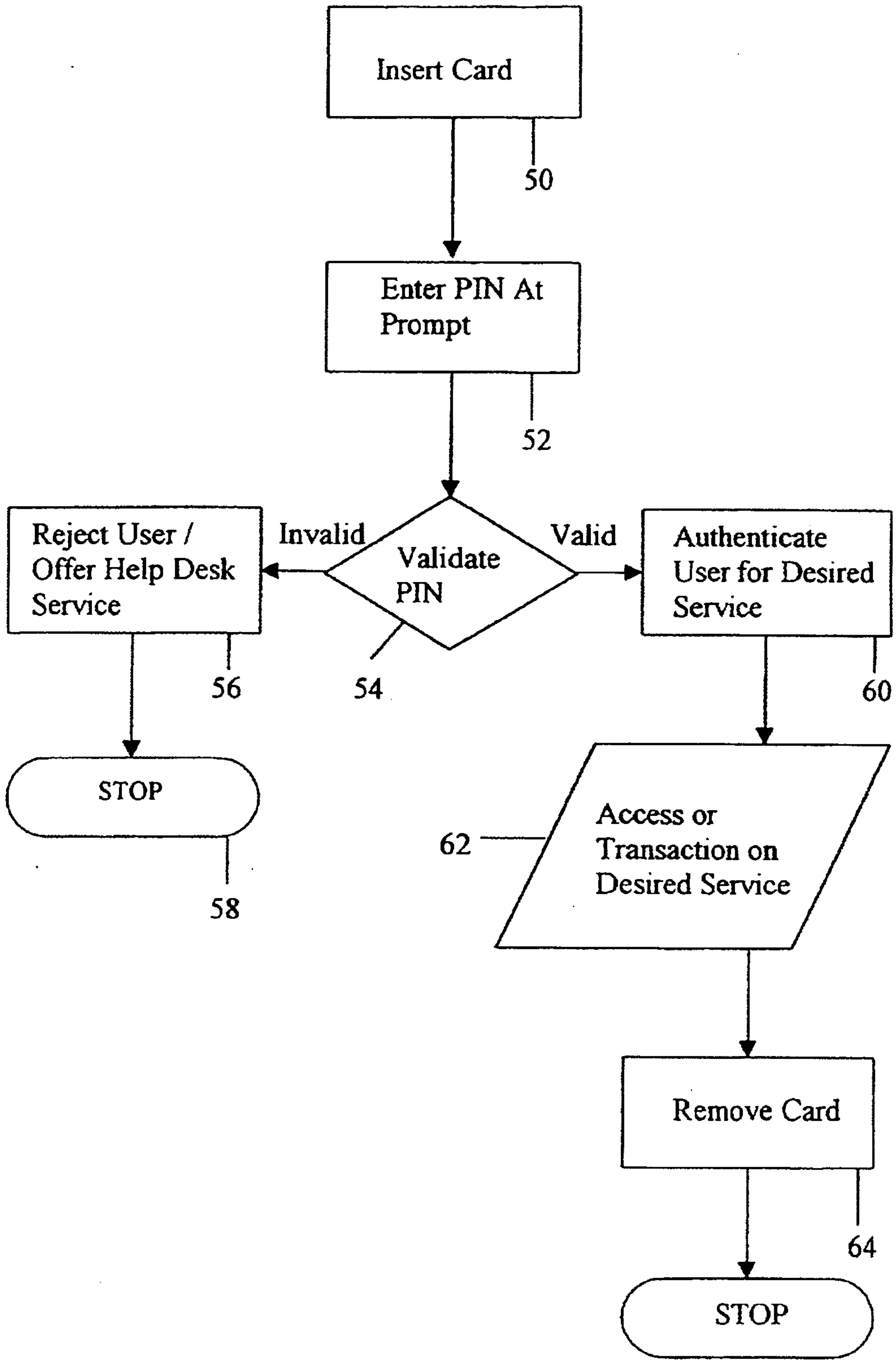

Insert Card
50
Enter PIN At Prompt
52
Invalid
Validate PIN
Valid
Reject User / Offer Help Desk Service
56
54
Authenticate User for Desired Service
60
STOP
58
62
Access or Transaction on Desired Service
Remove Card
64
STOP

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.

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 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 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 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 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.

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.

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 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 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 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.

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 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 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:

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.

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 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 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 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.

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 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 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 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 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 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.

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 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 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.

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 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, 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 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.

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 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 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 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 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.

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.

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.

8. A security system substantially as hereinbefore described with reference to and as illustrated in the accompanying drawings.

The Patent Office

10.

INVESTOR IN PEOPLE

Application No: GB 9921227.6 **Examiner:** Michael Logan
Claims searched: All **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 Cl (Ed.R): G4V (VAK); H4P (PDCSA)

Int Cl (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.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		

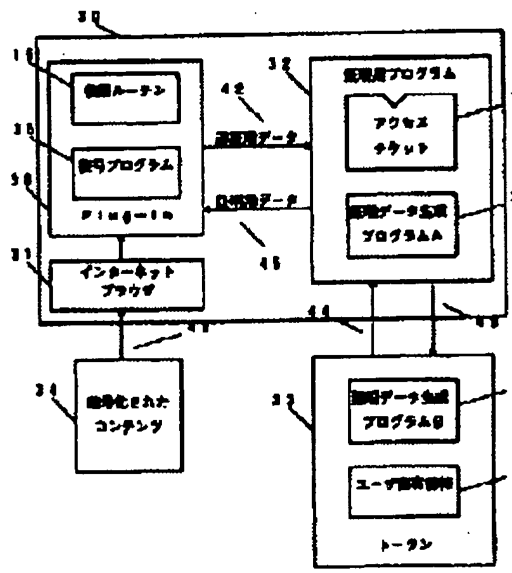
16) **Family number: 12389386 (JP11031130 A2)** | full-text | status | citations | < | > | ^ |

Title:	SERVICE PROVIDING DEVICE
Priority: Priority Map	JP19970184866 19970710

Family:	Publication number	Publication date	Application number	Application date	Link
Family Explorer	JP11031130 A2	19990202	JP19970184866	19970710	

Assignee(s):	FUJI XEROX CO LTD
Inventor(s):	KOJIMA SHUNICHI ; KONO KENJI ; NAKAGAKI JUHEI
International class (IPC 8):	G06F15/00 G09C1/00 H04L9/32 (Advanced/Invention); G06F15/00 G09C1/00 H04L9/32 (Core/Invention)
International class (IPC 1-7):	G06F15/00 G09C1/00 H04L9/32

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 A2)** | | | full-text | status | citations | < | > | ^ |

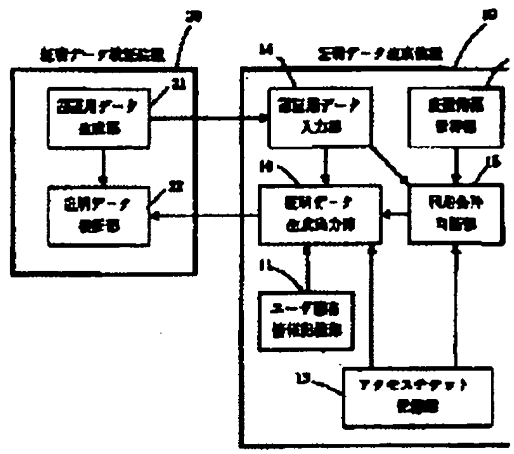
Title:	CERTIFICATION DATA GENERATING DEVICE
Priority: Priority Map	JP19970188801 19970714

Family: Family Explorer

Publication number	Publication date	Application number	Application date	Link
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: Family Explorer	Publication number	Publication date	Application number	Application date	Link
	JP11205306 A2	19990730	JP19980006267	19980116	

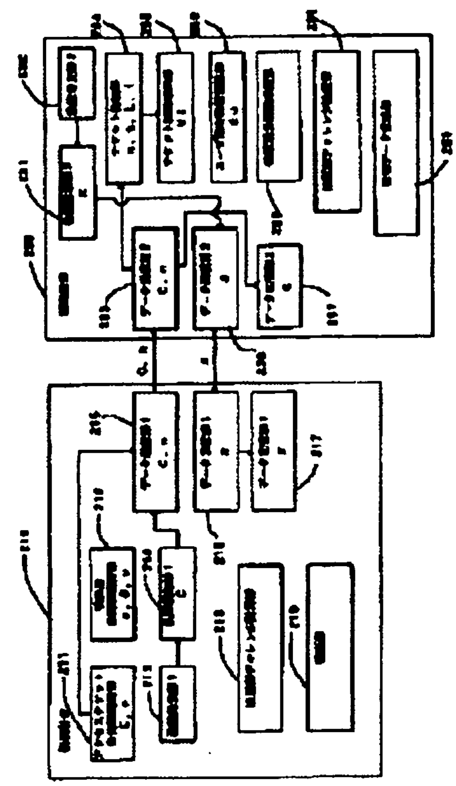
Assignee(s): FUJI XEROX CO LTD

Inventor(s): KOJIMA SHUNICHI ; KONO KENJI ; TAGUCHI MASAHIRO ; TERAO 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 authentification 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 authentification 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 authentification data (χ). The authentification device 250 receives the p and the (&epsi , v) to authenticate an authentication device open key based on input data and an authentication device open key identifier (v '), to authenticate the authentification device challenge and to revise contents of a ticket attribute record (VI) depending on the ticket attribute revision request (μ). Furthermore, an authentification 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**
Family Explorer	JP11215121 A2	19990806	JP19980016710	19980129	
	JP3791169 B2	20060628	JP19980016710	19980129	

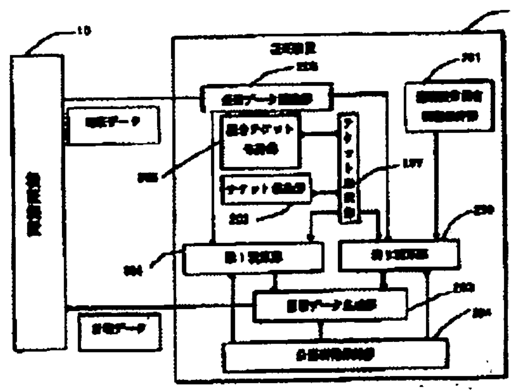
Assignee(s): FUJI XEROX CO LTD

Inventor(s): KIKO KENICHIROU

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: 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 RECORDI 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	

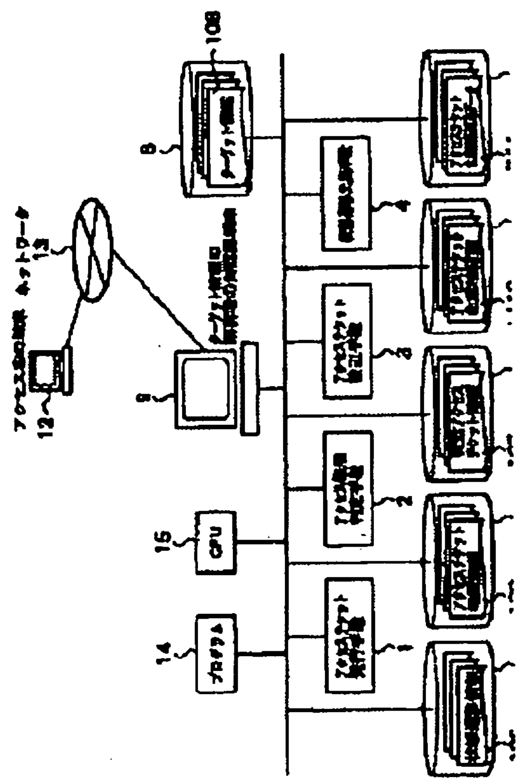
Assignee(s) (std): NIPPON TELEGRAPH AND TELEPHONE

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) | full-text | status | citations | < | > | ^ |
extended family

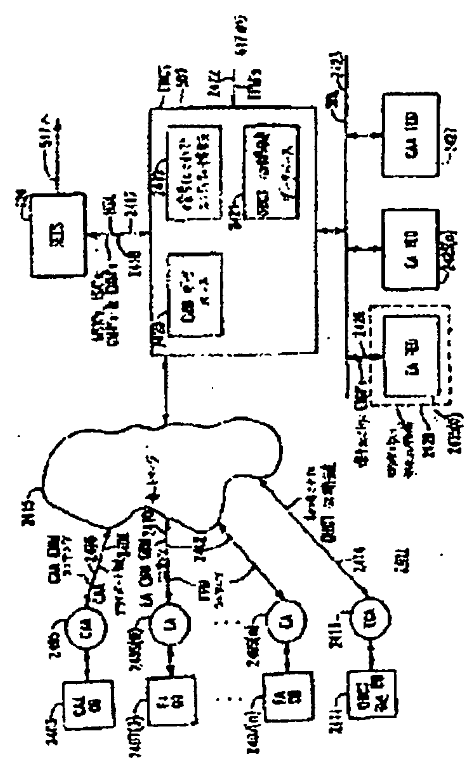
Title:	ENCRYPTION DEVICE USED IN A CONDITIONAL ACCESS SYSTEM
Priority: Priority Map	US19970054575P 19970801

Family: Family Explorer

Publication number	Publication date	Application number	Application date	Link
JP2005218143 A2	20050811	JP20050120426	20050418	
WO9907150 A1	19990211	WO1998US16145	19980731	

Assignee(s) (std):	SCIENTIFIC ATLANTA
Assignee(s):	SCIENTIFIC ATLANTA INC
Inventor(s) (std):	PALGON MICHAEL S ; PINDER HOWARD G
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 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):	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) | full-text | status | citations | < | > | ^ |
extended family

Title: CONDITIONAL ACCESS SYSTEM

Priority: US19970054575P 19970801 US19980126921 19980731
Priority Map

Family: Family Explorer

Publication number	Publication date	Application number	Application date	Link
JP2005253109 A2	20050915	JP20050120425	20050418	
WO9909743 A2	19990225	WO1998US16079	19980731	
WO9909743 A3	19990527	WO1998US16079	19980731	

Assignee(s) (std): SCIENTIFIC ATLANTA

Assignee(s): SCIENTIFIC ATLANTA INC

Inventor(s) (std): AKINS GLENDON L III ; PALGON MICHAEL S ; PINDER HOWARD G ; WASILEWSKI ANTHONY J

Inventor(s): AKINS GLENDON L

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 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

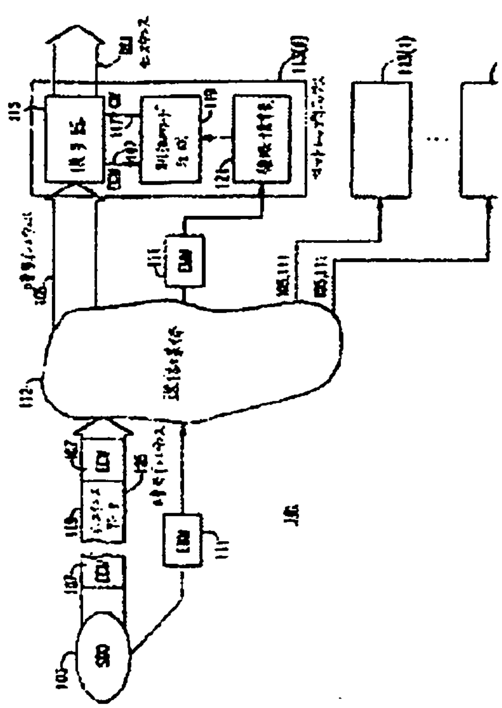
International class (IPC 8): H04H1/00 H04L9/08 H04N5/00 H04N7/16 H04N7/167 H04N7/173 (Advanced/Invention); H04H1/00 H04L9/08 H04N5/00 H04N7/16 H04N7/167 H04N7/173 (Core/Invention)

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

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 uncrypted 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.

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:
Family Explorer

Publication number	Publication date	Application number	Application date	Link
JP3185806 B2	20010711	JP19910352059	19911213	
JP5168039 A2	19930702	JP19910352059	19911213	

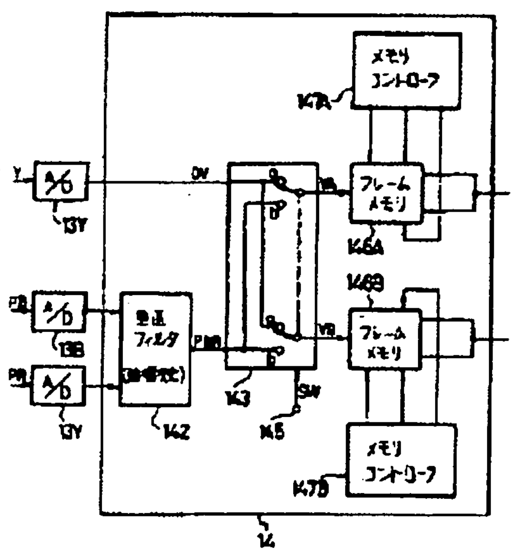
Assignee(s): SONY CORP

Inventor(s): ISHIMARU HIROYOSHI

International class (IPC 8): H04N11/22 H04N5/907 H04N9/80 H04N9/81 (Advanced/Invention); H04N11/06 H04N5/907 H04N9/80 H04N9/81 (Core/Invention)

International class (IPC 1-7): H04N11/22 H04N5/907 H04N9/80 H04N9/81

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.



PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G07F 7/08, 19/00, G06F 17/60 // 157:00	A1	(11) International Publication Number: WO 96/13814 (43) International Publication Date: 9 May 1996 (09.05.96)

(21) International Application Number: PCT/FI95/00591

(22) International Filing Date: 25 October 1995 (25.10.95)

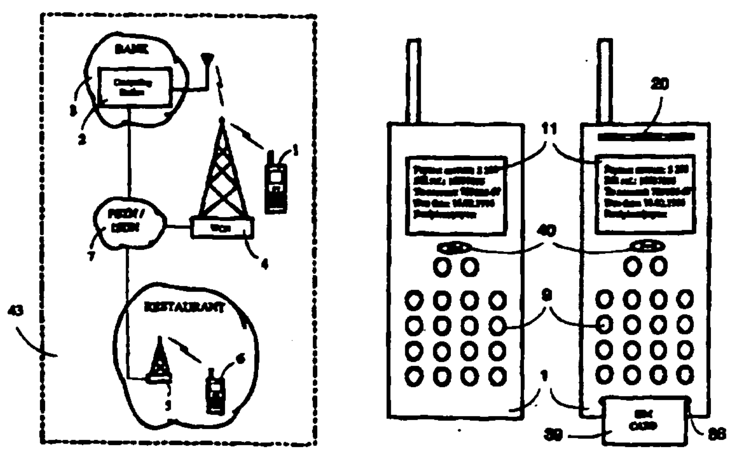
(30) Priority Data:
945075 28 October 1994 (28.10.94) FI

(71)(72) Applicant and Inventor: VAZVAN, Behruz [FI/FI]; Jämeräntaival 11 B 53, FIN-02150 Espoo (FI).

(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
With international search report.
With amended claims and statement.

(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	Kyrgystan	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 electronical 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** dose 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 form 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:

AMENDED SHEET (ARTICLE 19)

- 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

AMENDED SHEET (ARTICLE 19)

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

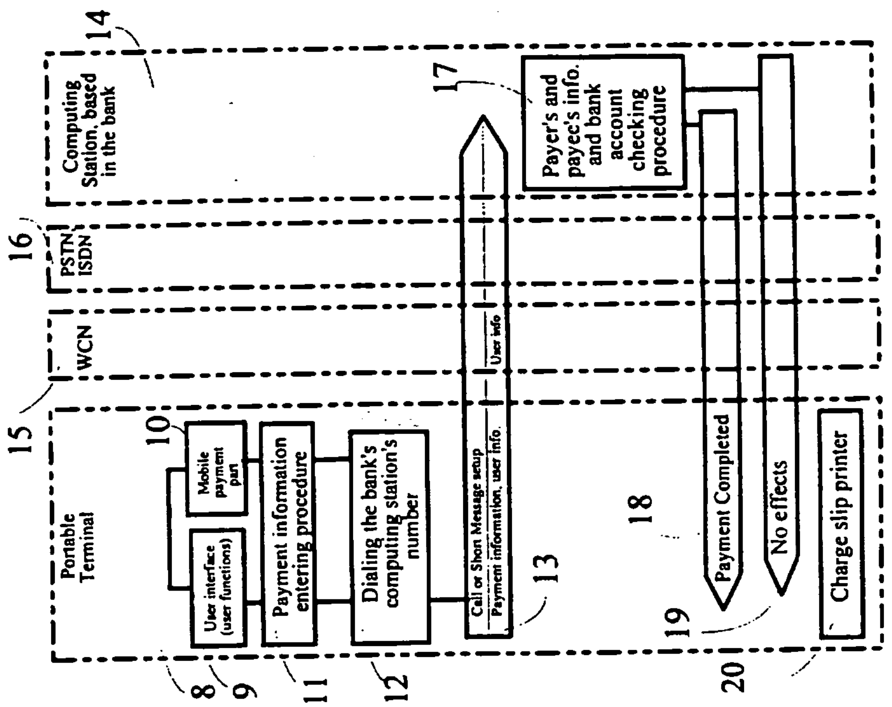

Portable Terminal
User interface (user functions)
Mobile payment part
Payment information entering procedure
Dialing the bank's computing station's number
Call or Short Message setup
Payment information, user info.
User info
WCN
PSTN ISDN
Computing Station, based in the bank
Payer's and payee's info. and bank account checking procedure
Payment Completed
No effects
Charge slip printer
8
9
10
11
12
13
14
15
16
17
18
19
20

FIG 1

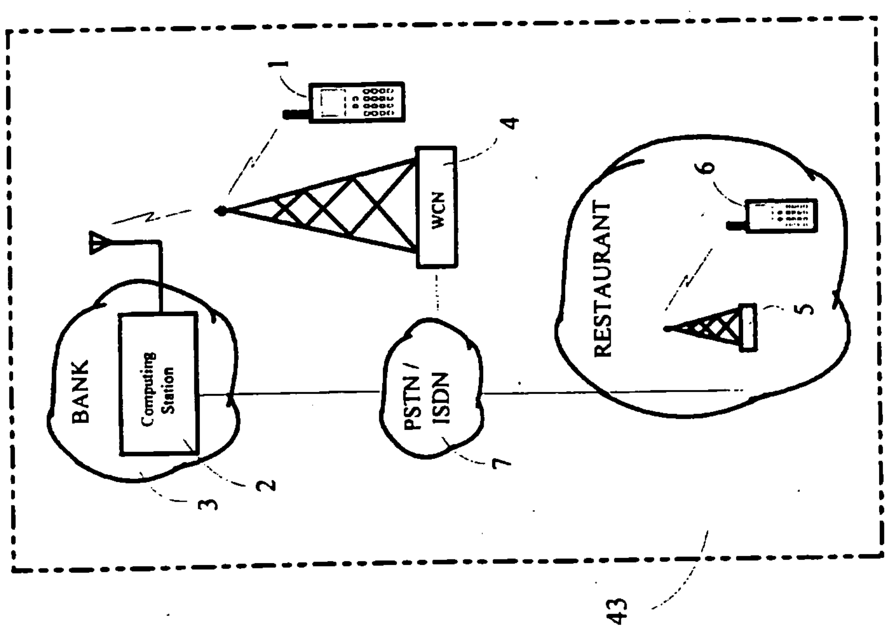
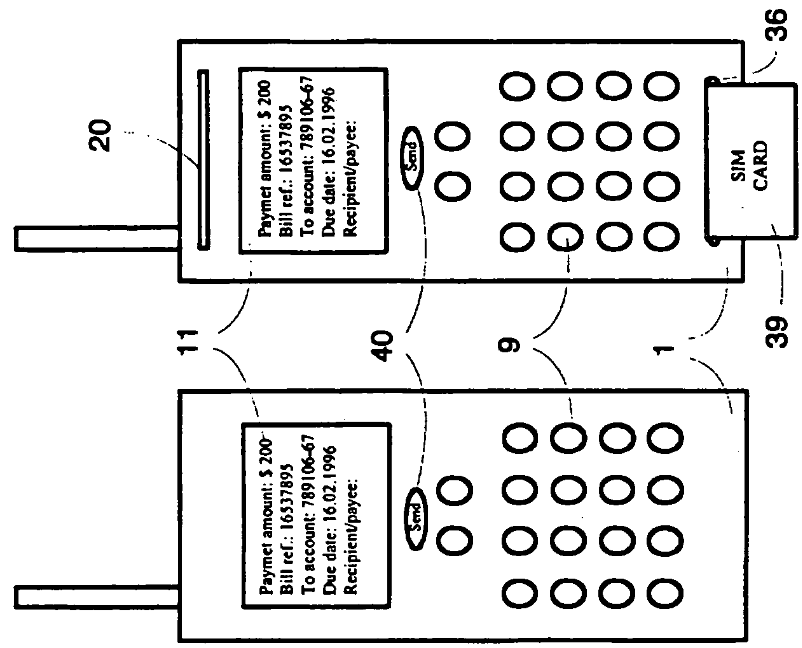
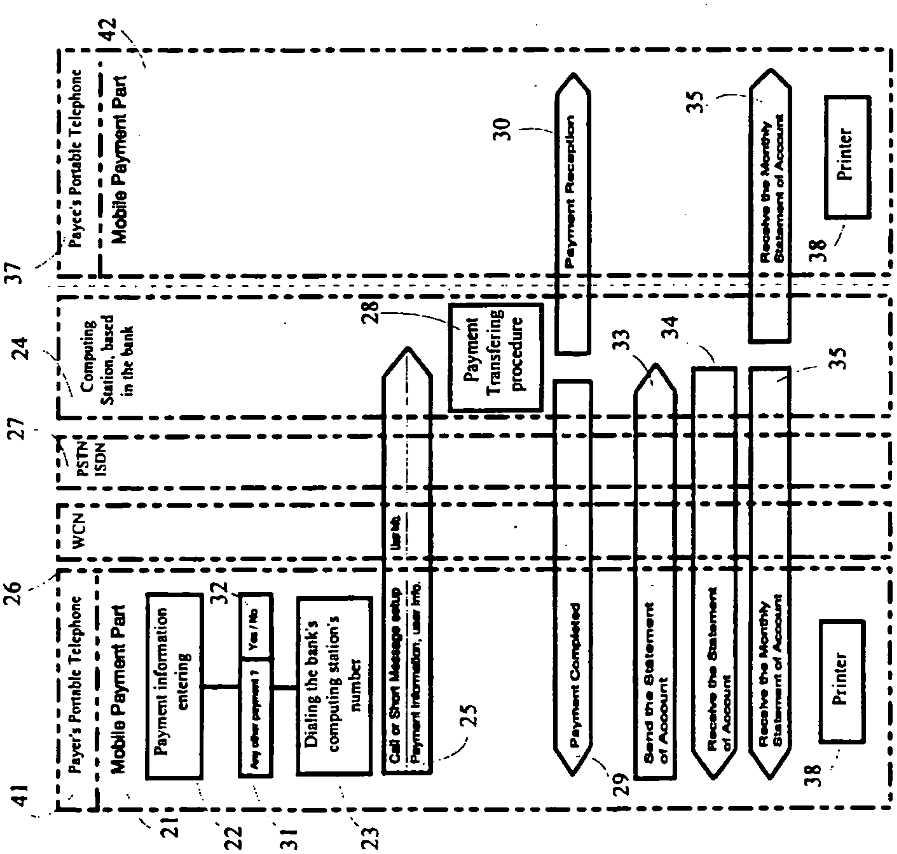

BANK
Computing Station
WCN
PSTN / ISDN
RESTAURANT
1
2
3
4
5
6
7
43

FIG 4



20
Paymet amount: $ 200
Bill ref.: 16537895
To account: 789106-67
Due date: 16.02.1996
Recipient/payee:
Send
SIM CARD
36
39
11
40
9
1

FIG 3



Payer's Portable Telephone
Mobile Payment Part
Payment information entering
Any other payment ?
Yes / No
Dialing the bank's computing station's number
Call or Short Message setup
Payment information, user info.
WCN
PSTN ISDN
Computing Station, based in the bank
Payment Transfering procedure
Payment Completed
Send the Statement of Account
Receive the Statement of Account
Receive the Monthly Statement of Account
Printer
Payee's Portable Telephone
Mobile Payment Part
Payment Reception
Receive the Monthly Statement of Account
Printer
41
21
22
31
23
26
32
25
29
38
27
24
28
33
34
35
37
42
30

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

☐ Further documents are listed in the continuation of Box C. ☒ See patent family 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" erlier 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	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

Form PCT/ISA/210 (second sheet) (July 1992)

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	

Form PCT/ISA/210 (patent family annex) (July 1992)

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04N 7/16, 7/167

A1

(11) International Publication Number: WO 00/46994

(43) International Publication Date: 10 August 2000 (10.08.00)

(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).

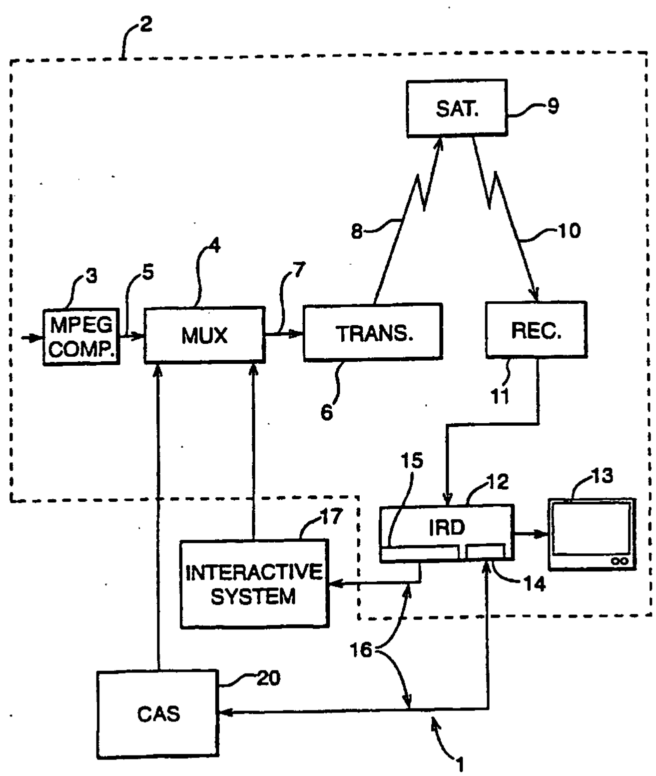
(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, 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).

Published
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.

(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 know 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.

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 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 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.

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 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.

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 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 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 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 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 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 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.

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.

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.

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.

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.

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 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 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.

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 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 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 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 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.

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.

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.

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.).

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.

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 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.

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.

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.

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".

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.

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:

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

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.

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.

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.

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.

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.

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.

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 sends, amongst other things, subscription rights to the daughter smartcard on request.

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.

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.

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

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.

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.

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.

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

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.

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.

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 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 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.

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.

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.

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.

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, 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 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).

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)

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.

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.

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.

The EMMs are passed to the Ciphering Unit (CU) 24 for ciphering 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.

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 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 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.

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 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 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.

<u>Data Encryption across an Interface</u>

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:

n	EMM (19 octets)	Key (8 octets)
1	EMM(1)	Key(1)
2	EMM(2)	Key(2)
3	EMM(3)	Key(3)
.	.	.
.	.	.
.	.	.
.	.	
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 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.

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:

Header	5 bytes
Encrypted key	10 bytes
Signature	9 bytes

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 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.

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 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.

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 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.

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.

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.

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. 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.

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. Authentification 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.

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.

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.

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 so on until the last decryption operation

64 on the final EMM message 65.

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.

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.

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.

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.

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.

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.

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.

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 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.

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, 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 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 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 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 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 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 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 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 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.

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.

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.

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 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.

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.

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. 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 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 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 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.

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.

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.

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.

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.

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.

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.

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.

FIG. 1

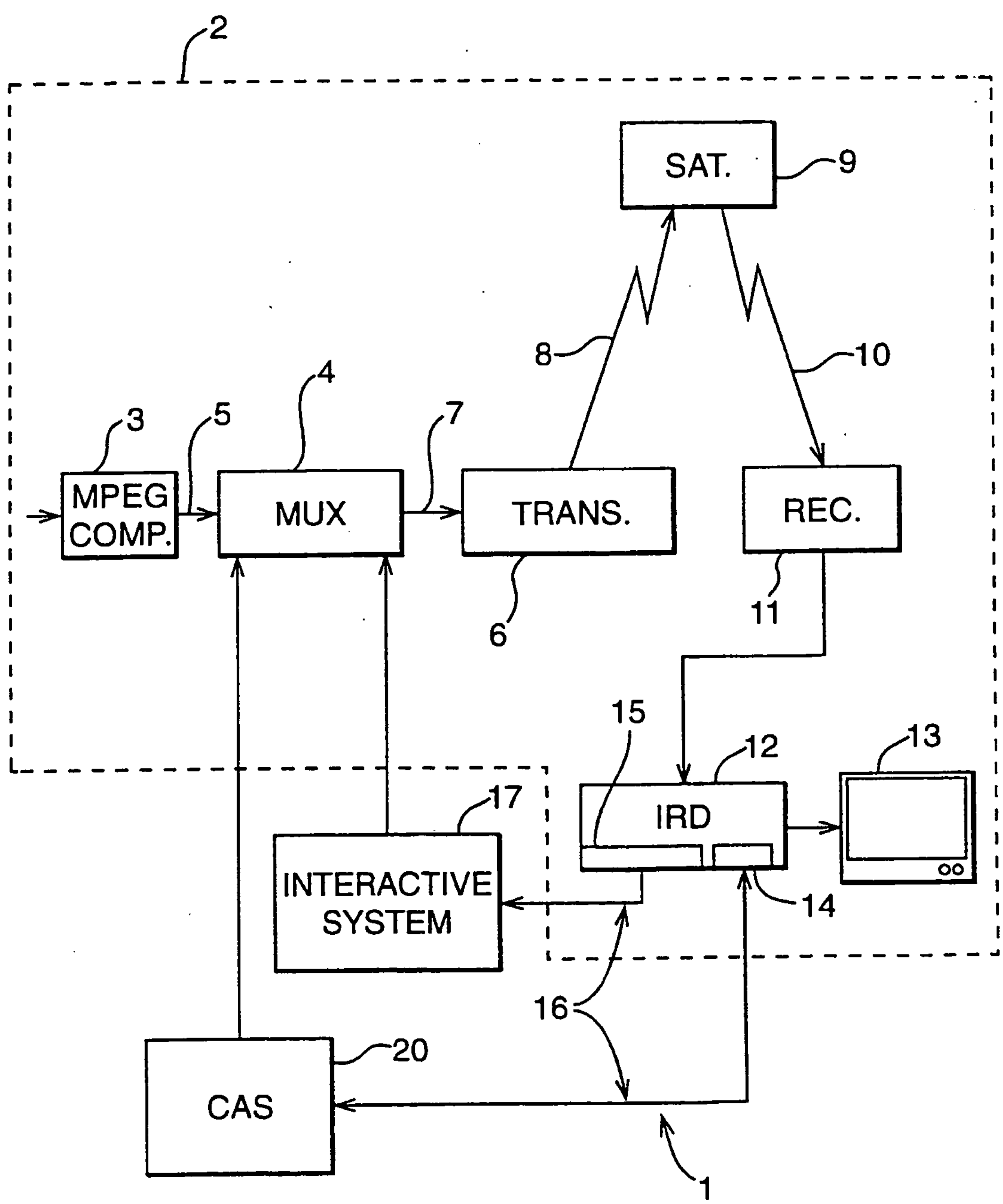
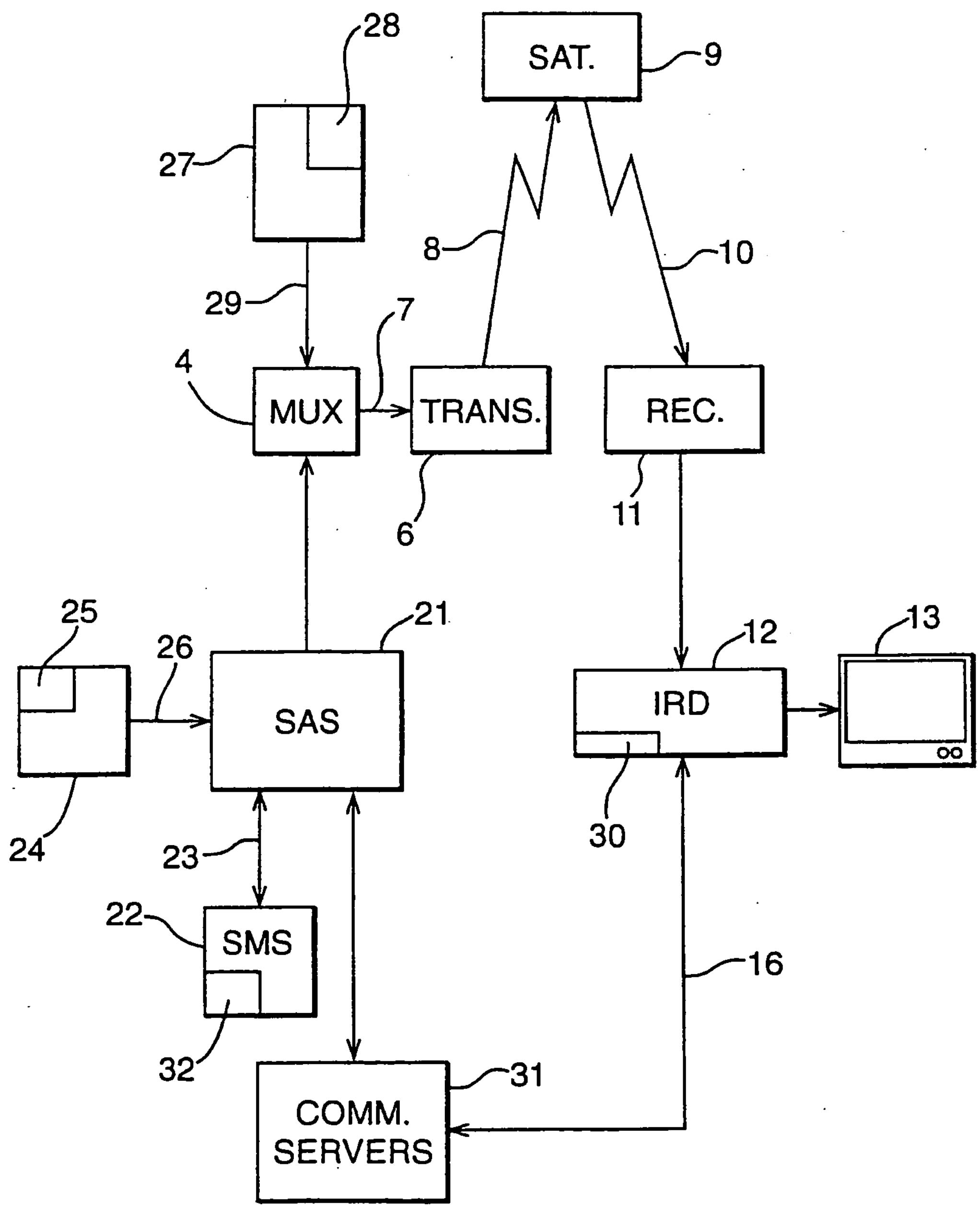

2
SAT.
9
8
10
4
3
5
7
MPEG COMP.
MUX
TRANS.
REC.
6
11
15
12
13
IRD
17
INTERACTIVE SYSTEM
14
16
20
CAS
1

FIG. 2


SAT.
9
28
27
8
10
29
7
4
MUX
TRANS.
REC.
6
11
25
26
21
SAS
12
IRD
13
24
23
30
22
SMS
16
32
31
COMM.
SERVERS

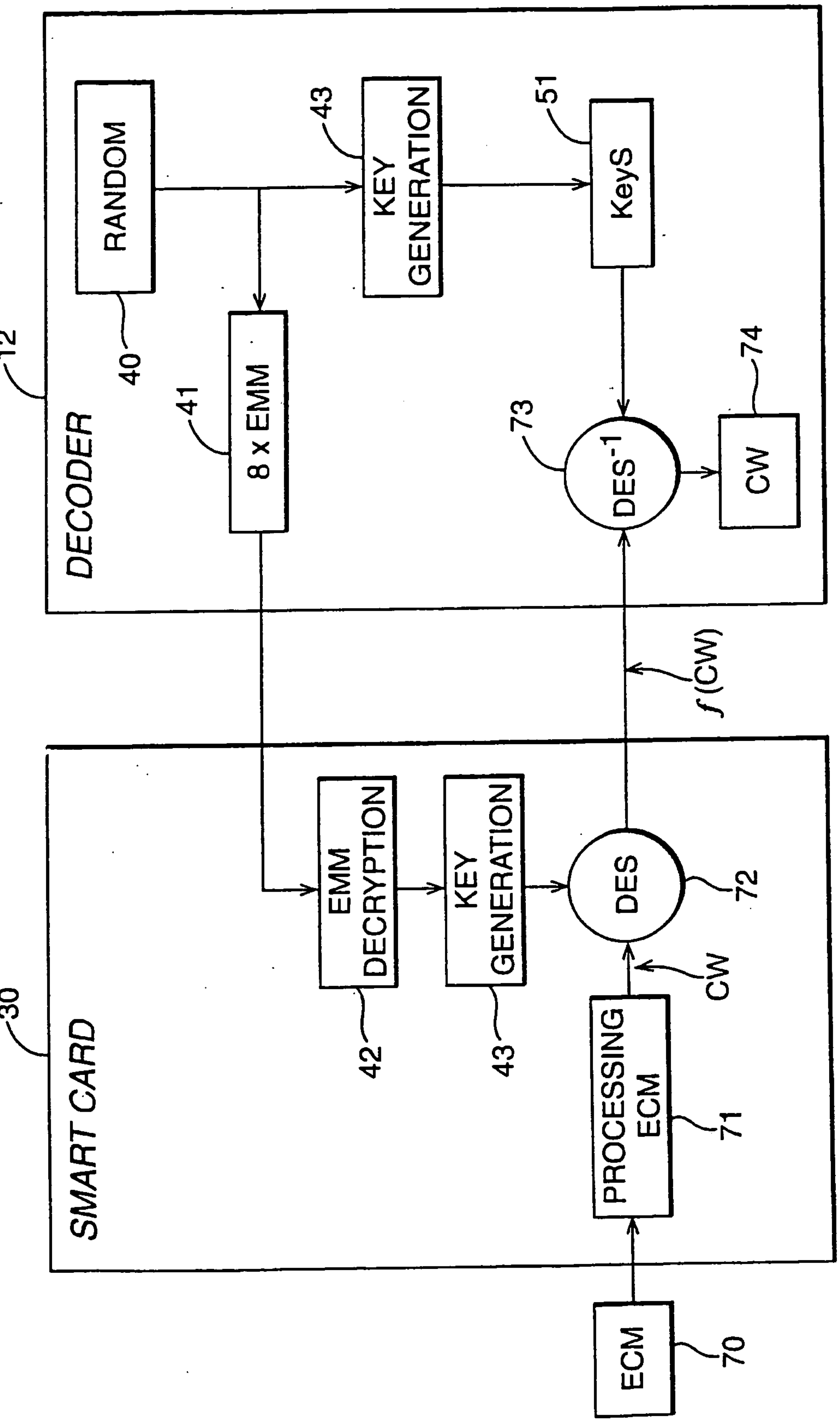

DECODER
12
RANDOM
40
KEY GENERATION
43
KeyS
51
8 x EMM
41
DES-1
73
CW
74
f(CW)
SMART CARD
30
EMM DECRYPTION
42
KEY GENERATION
43
DES
72
CW
PROCESSING ECM
71
ECM
70

FIG. 3

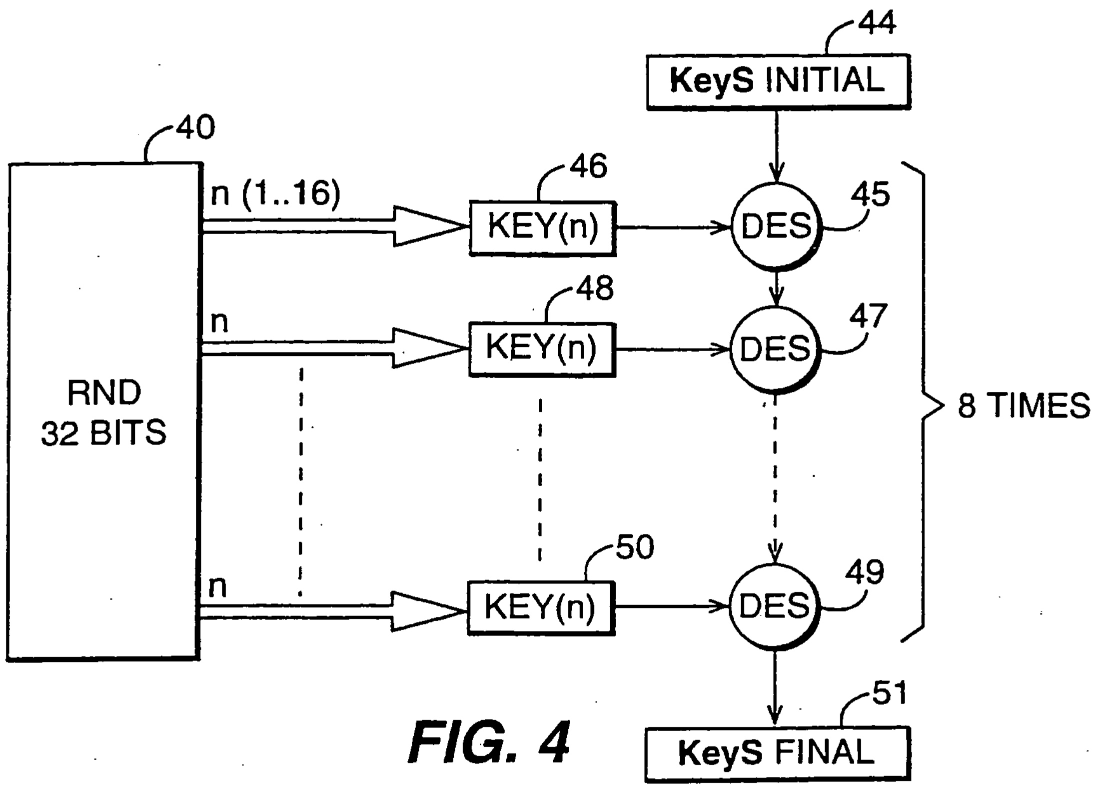

44
KeyS INITIAL
40
46
n (1..16)
KEY(n)
DES
45
48
n
KEY(n)
DES
47
RND
32 BITS
8 TIMES
50
n
KEY(n)
DES
49
51
KeyS FINAL

FIG. 4

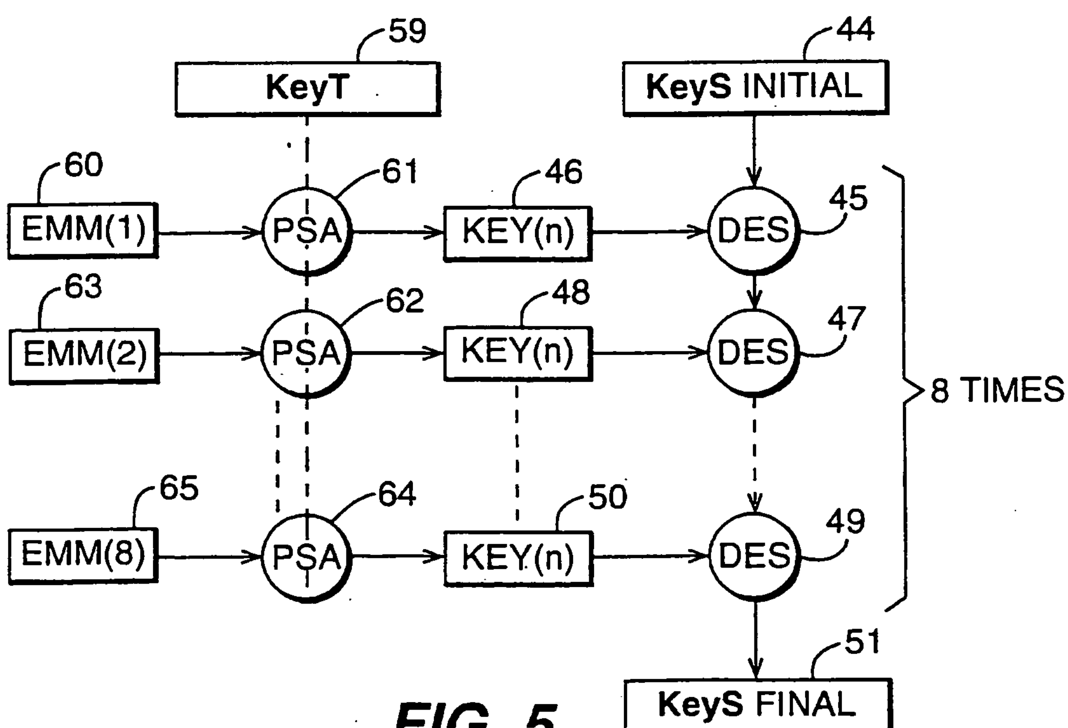

59
KeyT
44
KeyS INITIAL
60
EMM(1)
PSA
61
46
KEY(n)
DES
45
63
EMM(2)
PSA
62
48
KEY(n)
DES
47
8 TIMES
65
EMM(8)
PSA
64
50
KEY(n)
DES
49
51
KeyS FINAL

FIG. 5

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

-/--

[X] Further documents are listed in the continuation of box C.

[X] 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	Date of mailing of the international search report
31 May 2000	07/06/2000

Name and mailing address of the ISA	Authorized officer
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	Van der Zaal, R

1

Form PCT/ISA/210 (second sheet) (July 1992)

International 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	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	1-33

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

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)

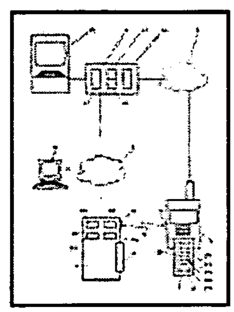
Applicants: **SWISSCOM MOBILE AG** [CH/CH]; Schwarztorstrasse 61 CH-3050 Bern (CH) *(All Except US)*.
RITTER, Rudolf [CH/CH]; Rossweidweg 8 CH-3052 Zollikofen (CH) *(US Only)*.
LAUPER, Eric [CH/CH]; Hochfeldstrasse 96 CH-3012 Bern (CH) *(US Only)*.

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)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(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 Classification[7]: 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).

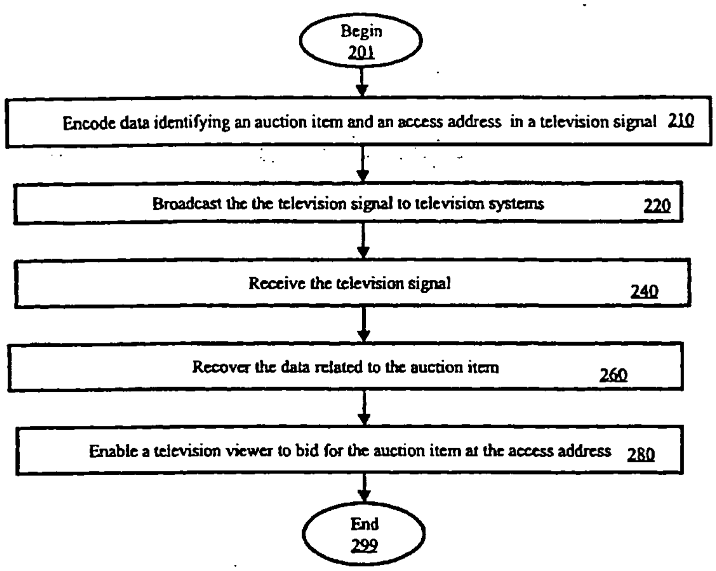
(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, 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, "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 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 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 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 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 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 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 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.

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.

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 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 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 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

In one embodiment, a highest present bid may be encoded in the television signal, and the user may submit a higher bid that 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.

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.

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.

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 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 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.

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 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 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 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 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 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

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 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 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, 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 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 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 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

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. A method of enabling a viewer of a television system to participate in auctions, said method comprising:

(a) encoding in a television signal a data describing an auction item and an access address of a server at which auction service for said auction item is provided; and

(b) transmitting said television signal,

wherein said data can be used to enable said viewer to bid for said auction item at said server.

2. The method of claim 1, wherein said method further comprises:

(c) receiving said television signal encoded with said data in a transaction enabler;

(d) recovering said data encoded in said television signal;

(e) displaying information describing said auction item on said television system;

(f) enabling said viewer to bid at said server specified by said access address.

3. The method of claim 2, further comprising:

(g) enabling said viewer to specify a bid price for said auction item.

4. The method of claim 3, wherein said enabling said viewer to specify said bid price comprises:

(h) enabling said viewer to indicate said bid price; and

(i) transmitting said bid price to said server at said access address.

5. The method of claim 4, wherein said access address comprises a telephone number

of said server, and said method further comprises:

(j) encoding a unique code identifying said auction item;

(k) recovering said unique code in said transaction enabler; and

(l) transmitting said unique code along with said bid price to said server,

whereby said server can easily associate said bid price with said auction item using said unique code.

6. The method of claim 4, wherein said access address comprises a universal resource locator (URL) of a web site, wherein said web site comprises said server, and wherein steps (h) and (i) comprise the further step of enabling said viewer to indicate said price on a web page provided by said web site.

7. The method of claim 1, further comprising:

(m) encoding a present highest bid in said television signal, wherein said present highest bid may be displayed to said viewer before said viewer decides to submit a bid.

8. The method of claim 7, wherein said server comprises a web site, and said method comprising the further step of retrieving said present highest bid from said web site.

9. The method of claim 1, wherein step (a) comprises the step of encoding said data in non-display portion of said television signal.

10. The method of claim 1, wherein step (a) comprises the further step of encoding said data in a non-display portion of said television signal.

11. The method of claim 10, wherein said non-display portion comprises vertical blanking interval (VBI).

12. The method of claim 1, further comprising:

transmitting an updated highest bid price in said television signal, wherein said updated highest bid price corresponds to a present highest bid for said auction item.

13. The method of claim 12, further comprising:

retrieving said updated bid price from said server,

wherein said step of transmitting said updated highest bid price is performed after said step of retrieving said updated bid price from said server.

14. The method of claim 13, further comprising:

enabling said viewer to request a bid history; and

displaying all of said updated bid prices to said viewer.

15. The method of claim 14, wherein said display corresponding to said bid history further comprises a description of the bidder corresponding to each of said present highest bid.

16. The method of claim 1, wherein said data further comprises a time at which auction for said auction item closes.

17. A method of enabling a viewer of a television system to participate in auctions,

said method comprising:

(a) receiving in a transaction enabler a television signal encoded with a data, said data including a description of an auction item and an access address of a server at which auction service for said auction item is provided;

(b) recovering said data encoded in said television signal;

(c) displaying said description of said auction item on said television system;

(d) enabling said viewer to bid at said server specified by said access address.

18. The method of claim 17, further comprising:

(e) enabling said viewer to indicate said bid price; and

(f) transmitting said bid price to said server at said access address.

19. The method of claim 4, wherein said access address comprises a telephone number of said server, and said method further comprises:

(g) encoding a unique code identifying said auction item;

(h) recovering said unique code in said transaction enabler; and

(i) transmitting said unique code along with said bid price to said server,

whereby said server can easily associate said bid price with said auction item using said said unique code.

20. An environment enabling a viewer of a television system to participate in auctions, said environment comprising:

encoding means for encoding in a television signal a data describing an auction item and an access address of a server at which auction service for said auction item is provided; and

transmission means for transmitting said television signal,

wherein said data can be used to enable said viewer to bid for said auction item at said server.

21. An environment enabling a viewer of a television system to participate in auctions, said environment comprising:

receiving means for receiving a television signal encoded with a data, said data including a description of an auction item and an access address of a server at which auction service for said auction item is provided;

recovery means for recovering said data encoded in said television signal;

displaying means for displaying said description of said auction item on said television system;

enabling means for enabling said viewer to bid at said server specified by said access address.

22. An environment enabling a viewer of a television system to participate in auctions, said environment comprising:

a broadcast system to encode in a television signal a data describing an auction item and an access address of a server at which auction service for said auction item is provided, said broadcast system being designed also to transmit said television signal,

wherein said data can be used to enable said viewer to bid for said auction item at said server.

23. The environment of claim 22, wherein said broadcast system comprises:

a production block to generate images to encode in a display data portion of said television signal;

an authoring block to encode said data in said television signal; and

a broadcast block to transmit said television signal containing said images and said data.

24. The environment of claim 23, further comprising an auction data interface to receive a present highest bid from a server, said auction data interface to provide said present highest bid to said authoring block, wherein said authoring block encodes said present highest bid in said television signal.

25. The environment of claim 24, further comprising a timing determination block to determine the time at which said authoring block encodes said data including said present highest bid in said television signal.

26. The environment of claim 22, further comprising:

a viewer bidding system to receive said television signal, and enabling said viewer to submit a bid and participate in said auction.

27. The environment of claim 26, wherein said viewer bidding system comprises:

a television system;

a remote control which enables said viewer to submit said bid; and a transaction enabler coupled to said television system and to receive said commands from said remote control, said transaction enabler to recover said data encoded in said television signal and display information contained in said data on said television,

wherein said viewer can submit said bid using said remote control.

28. The environment of claim 27, wherein said transaction enabler is integrated within said television system.

29. The environment of claim 27, wherein said transaction enabler is provided external to said television system, and wherein said transaction enabler overlays a window with information contained in said data on images encoded in the display data of said television signal.

30. The environment of claim 27, wherein said window is displayed in a transparent mode on said images.

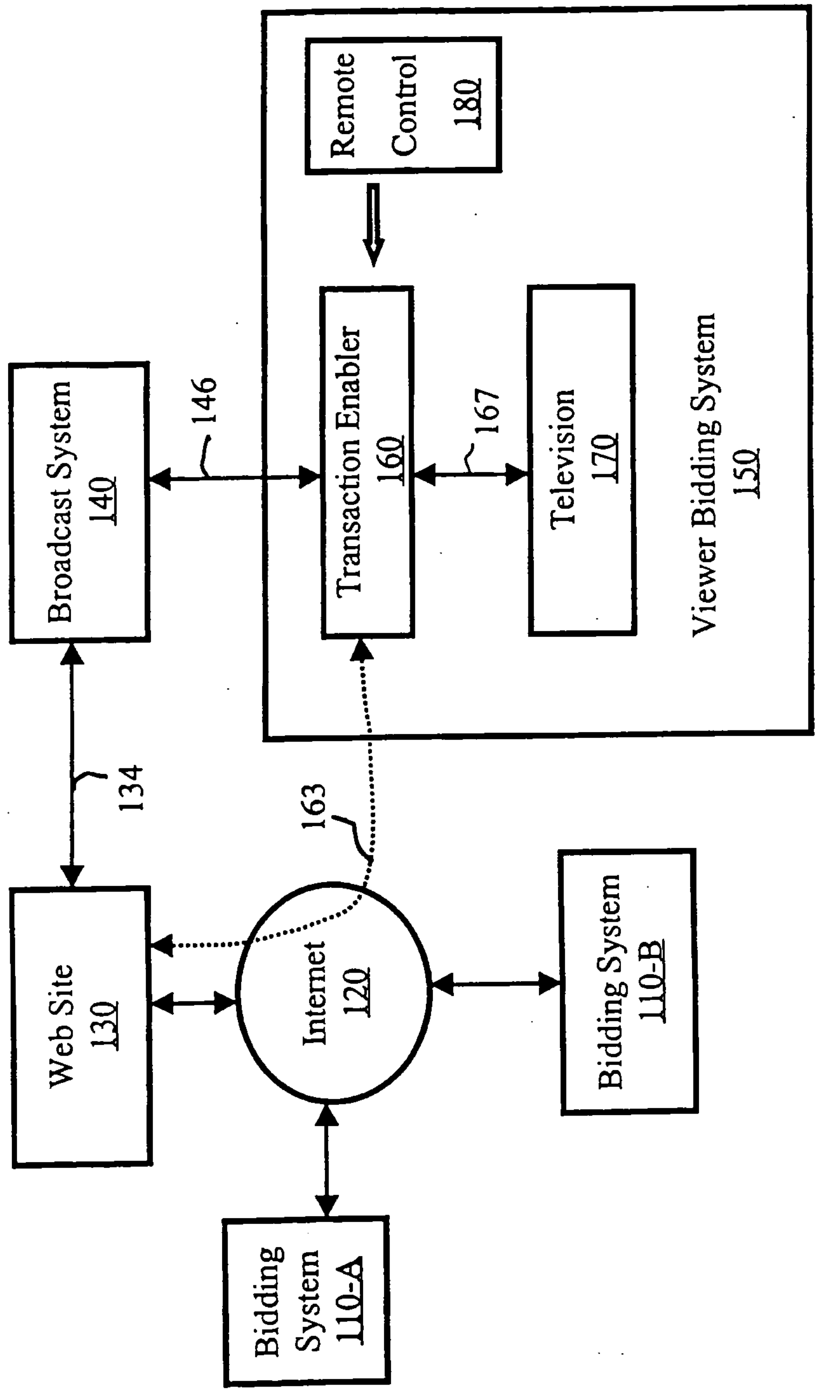

Remote Control 180
Transaction Enabler 160
Television 170
Viewer Bidding System 150
Broadcast System 140
146
167
134
163
Web Site 130
Internet 120
Bidding System 110-B
Bidding System 110-A

Figure 1

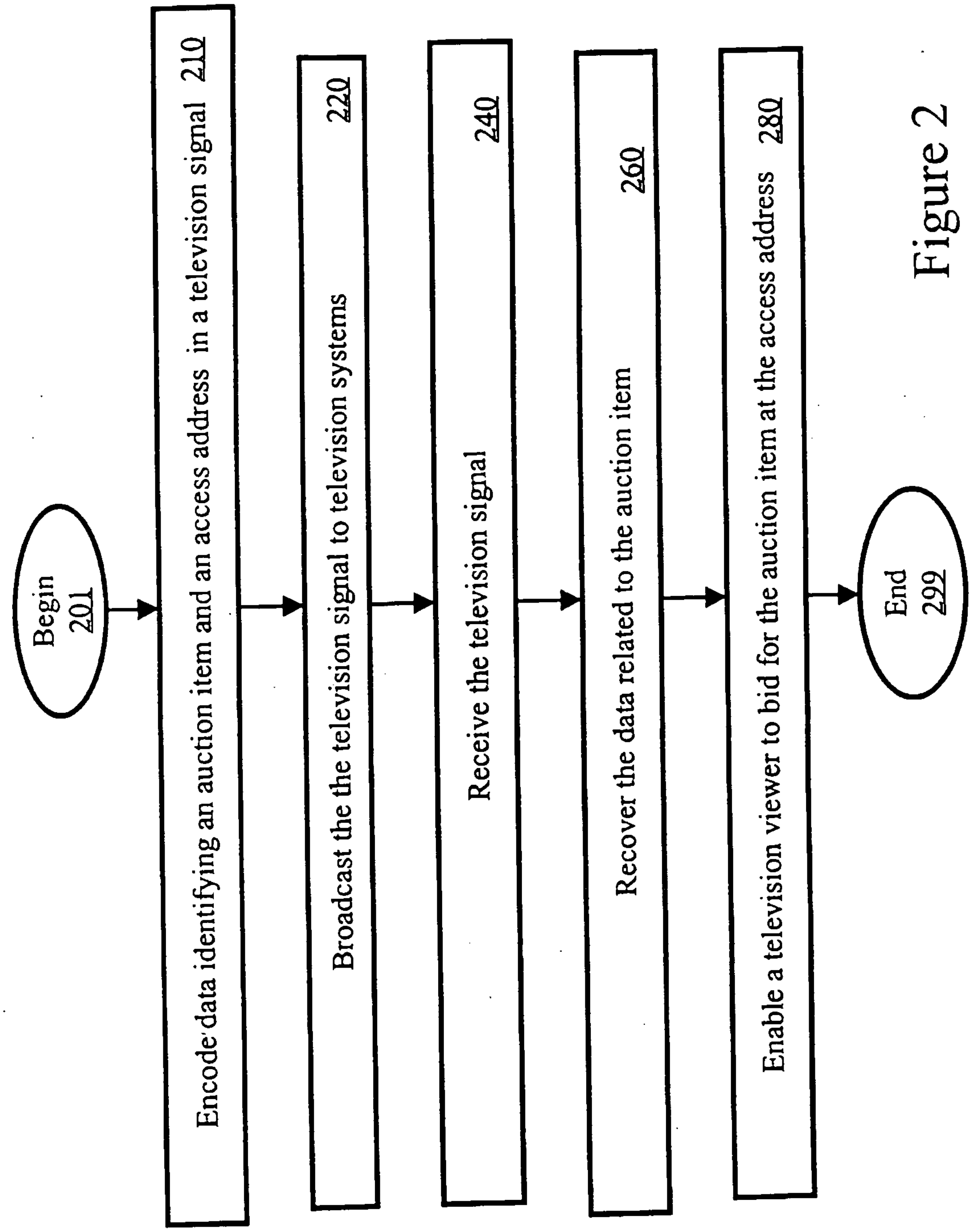

Begin 201
Encode data identifying an auction item and an access address in a television signal 210
Broadcast the the television signal to television systems 220
Receive the television signal 240
Recover the data related to the auction item 260
Enable a television viewer to bid for the auction item at the access address 280
End 299

Figure 2

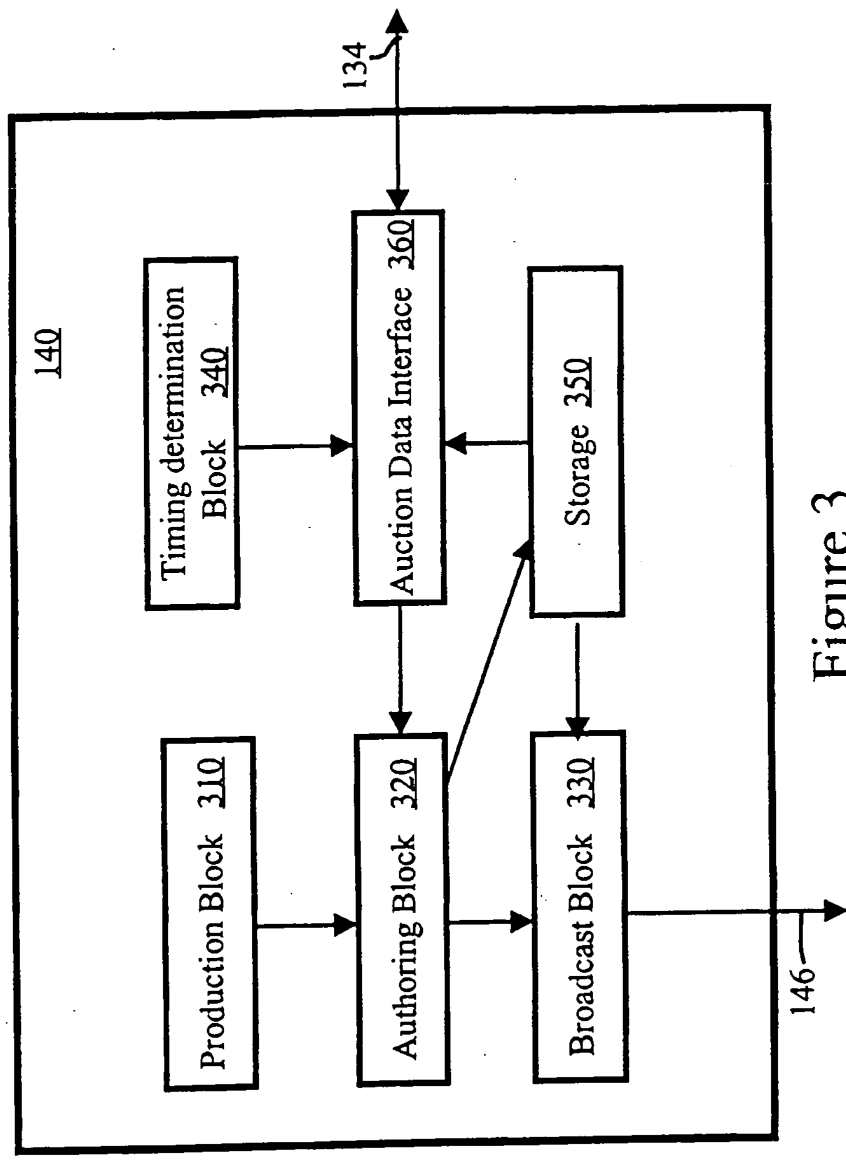

134
140
Timing determination Block 340
Auction Data Interface 360
Storage 350
Production Block 310
Authoring Block 320
Broadcast Block 330
146

Figure 3

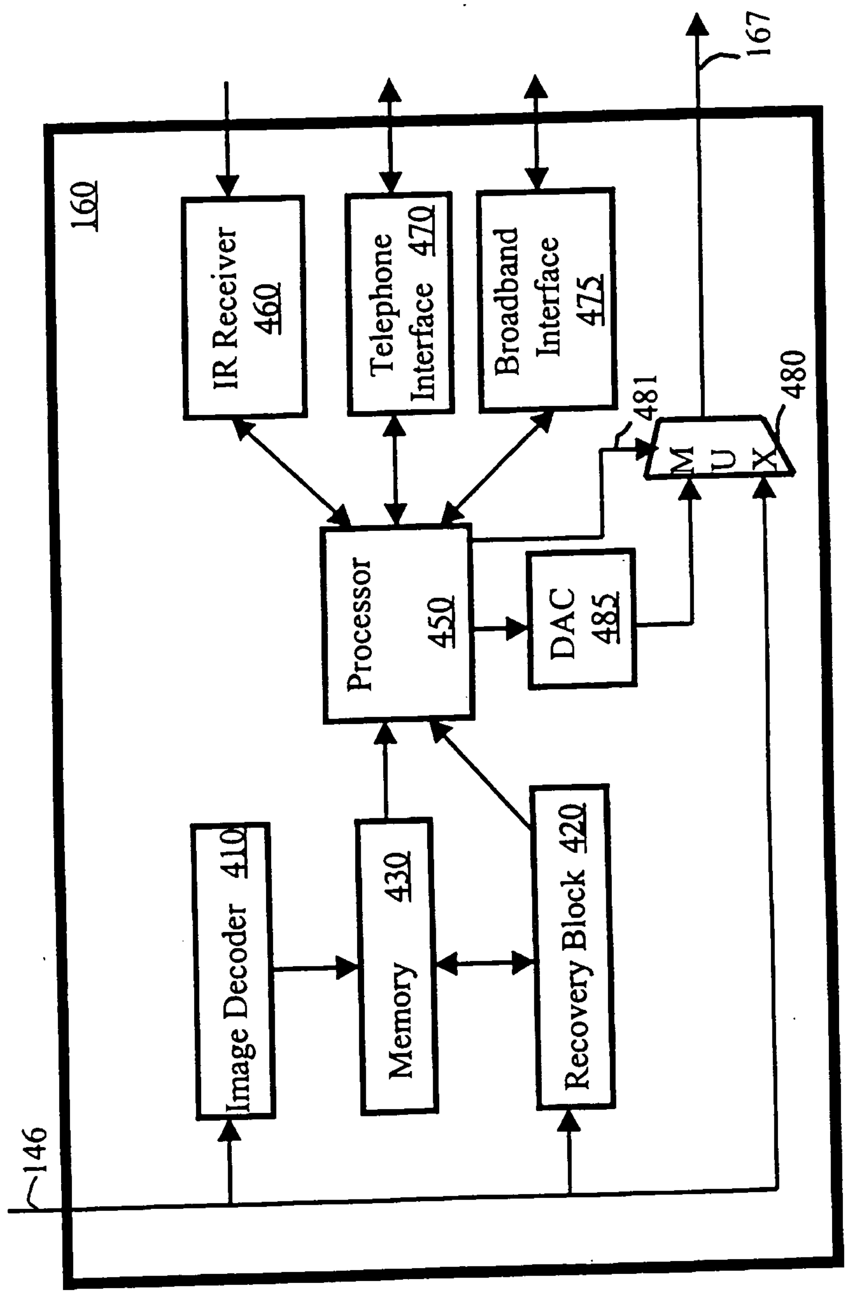

160
IR Receiver 460
Telephone Interface 470
Broadband Interface 475
Processor 450
DAC 485
MUX
481
480
167
Image Decoder 410
Memory 430
Recovery Block 420
146

Figure 4

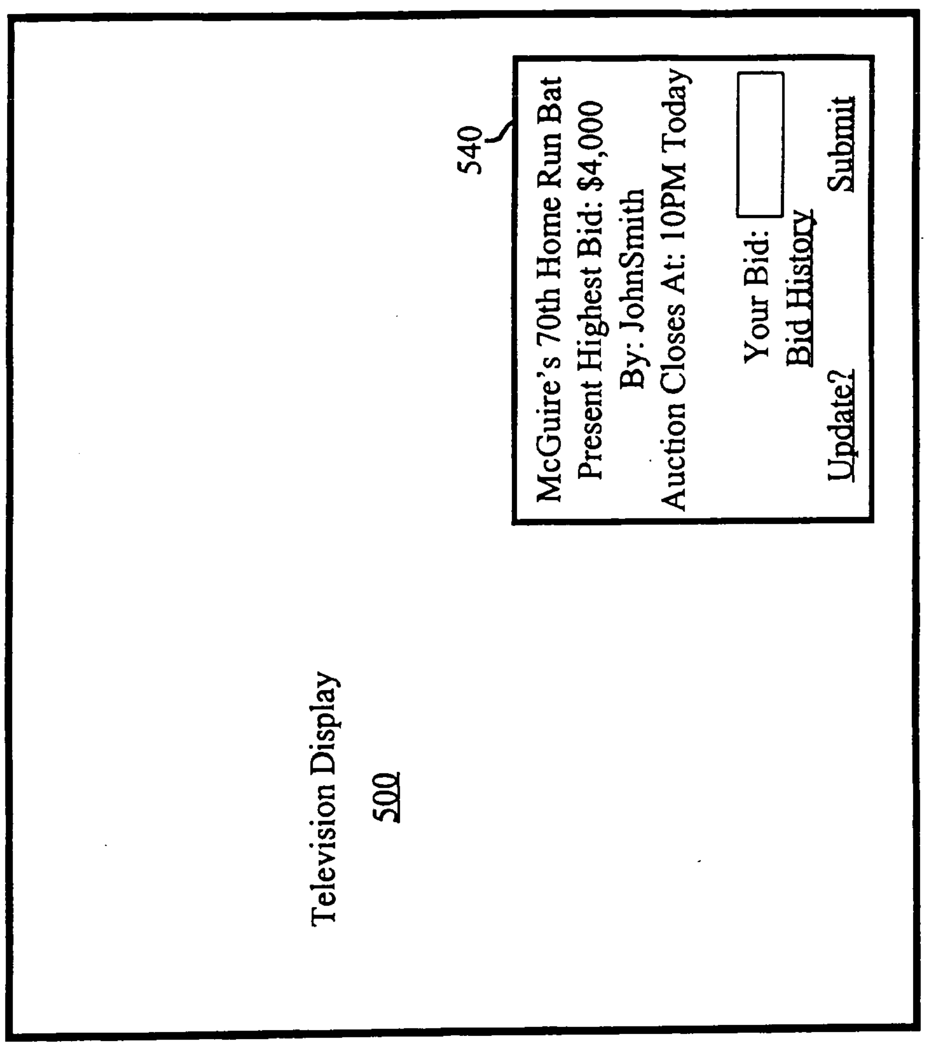

Television Display
500
540
McGuire's 70th Home Run Bat
Present Highest Bid: $4,000
By: JohnSmith
Auction Closes At: 10PM Today
Your Bid:
Bid History
Update?
Submit

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

[X] Further documents are listed in the continuation of Box C. [ ] 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	
"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	
"P" document published prior to the international filing date but later than the priority date claimed	"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
22 AUGUST 2000	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 Eugenia Logan Telephone No. (703) 305-9760

Form PCT/ISA/210 (second sheet) (July 1998)*

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

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

Search result: 1 of 1

(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)*.

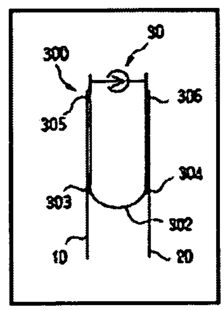
Inventor: **BOIS, Henri, Georges** [FR/FR]; 61, boulevard d'Inkermann, F-92200 Neuilly sur Seine (FR).

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)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(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 Classification[7]: 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).

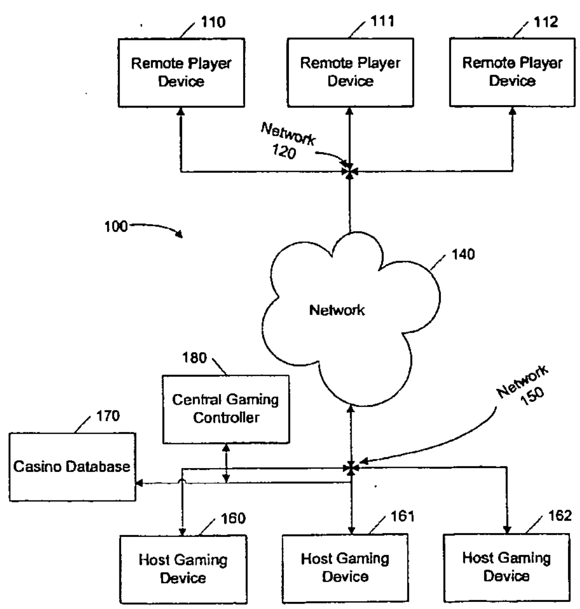
(81) Designated States (*national*): AE, AG, AL, AM, AT (utility model), AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (utility model), CZ, DE (utility model), DE, DK (utility model), DK, DM, DZ, EC, EE (utility model), EE, EG, ES, FI (utility model), FI, GB, GD, GE, GH, GM, HR, HU, 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 be 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 mèthod 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.

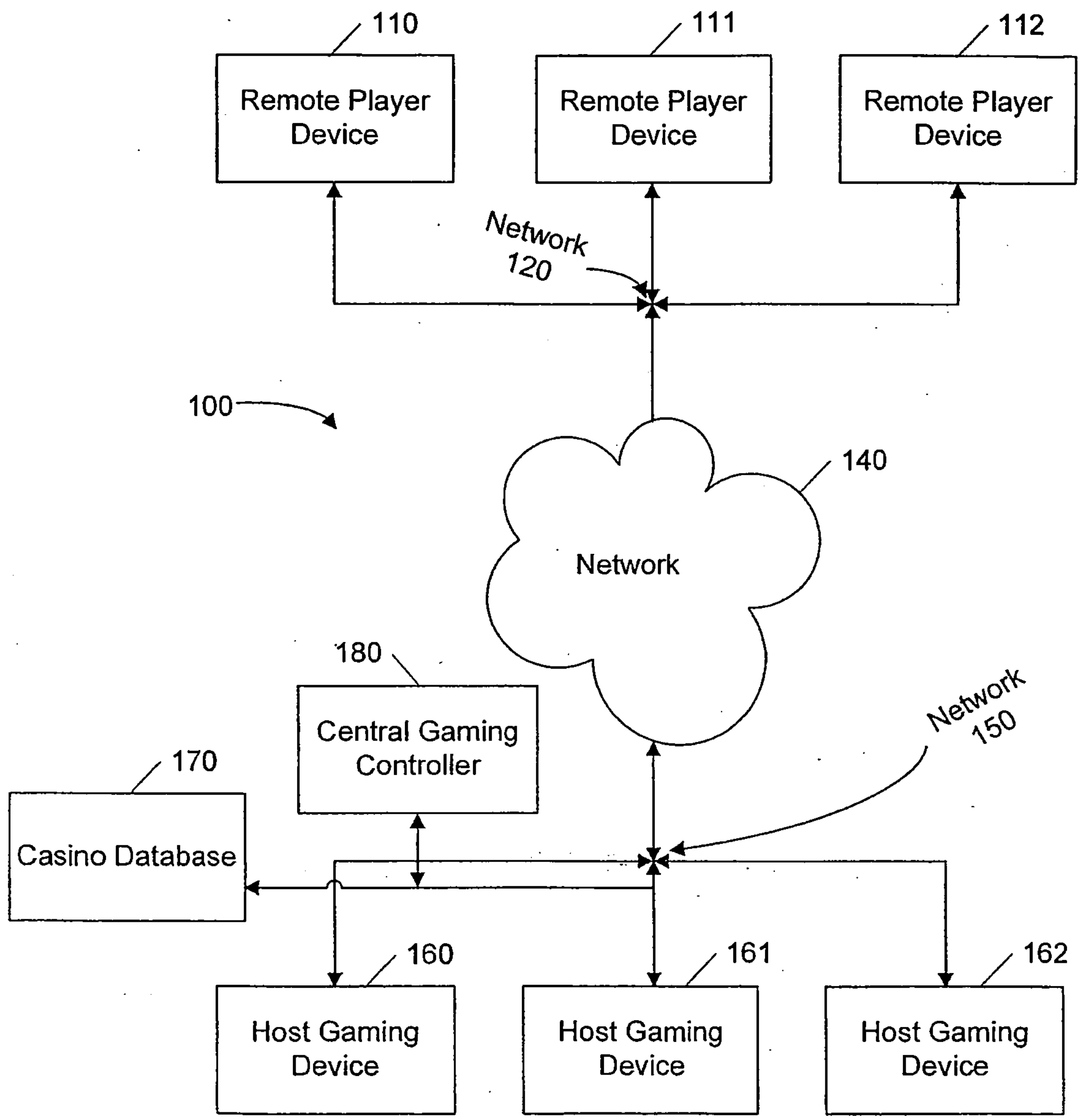
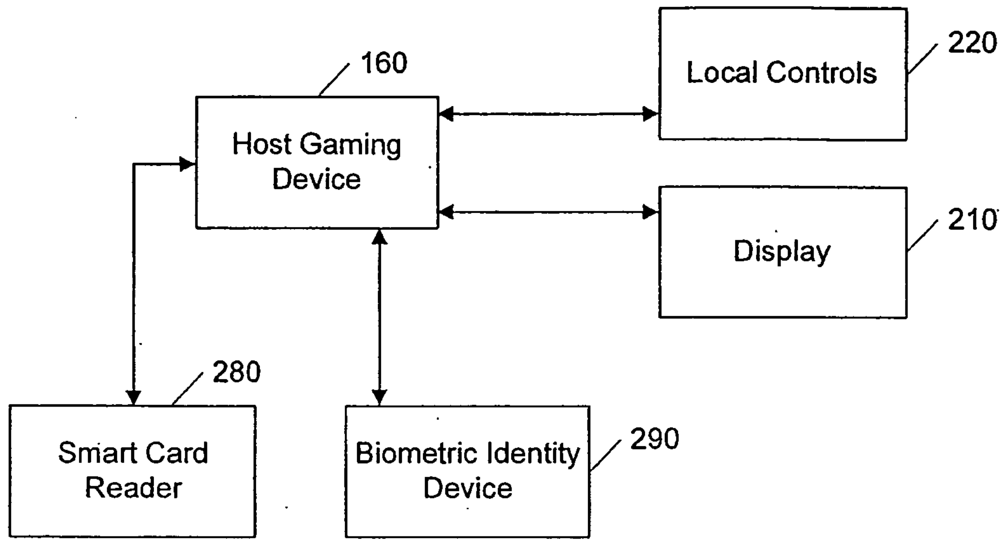

110
Remote Player Device
111
Remote Player Device
112
Remote Player Device
Network 120
100
140
Network
180
Central Gaming Controller
Network 150
170
Casino Database
160
Host Gaming Device
161
Host Gaming Device
162
Host Gaming Device

FIG. 1



160
Host Gaming Device
Local Controls
220
Display
210
280
Smart Card Reader
Biometric Identity Device
290

FIG. 2

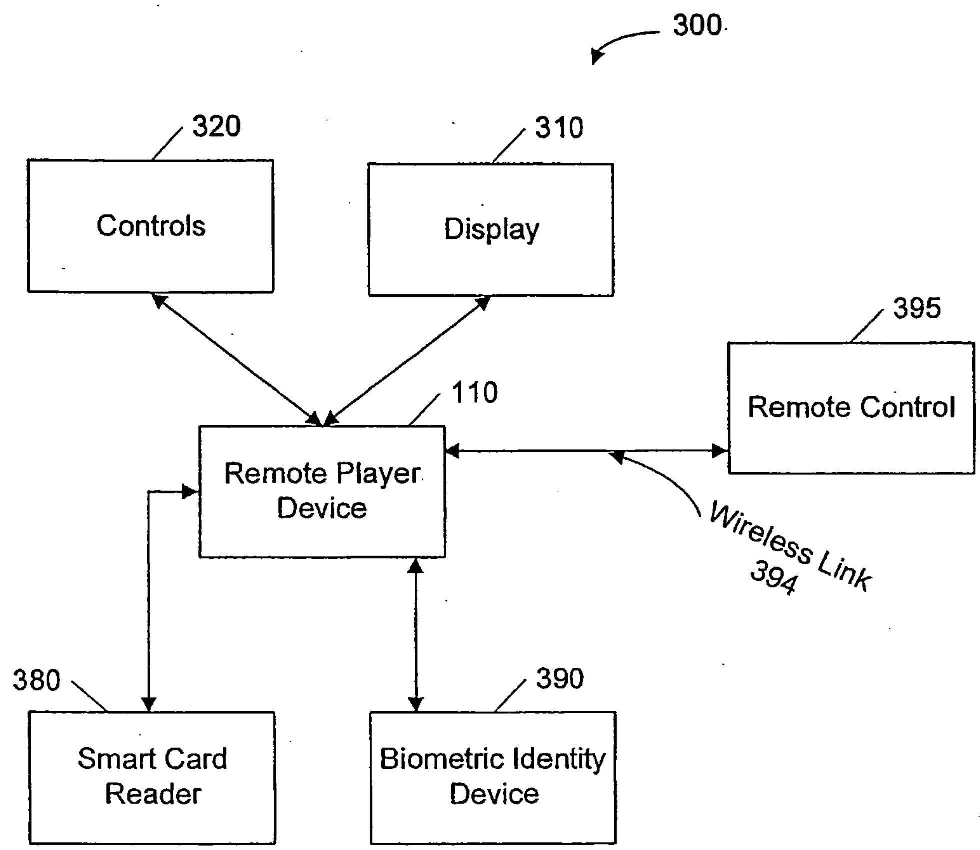
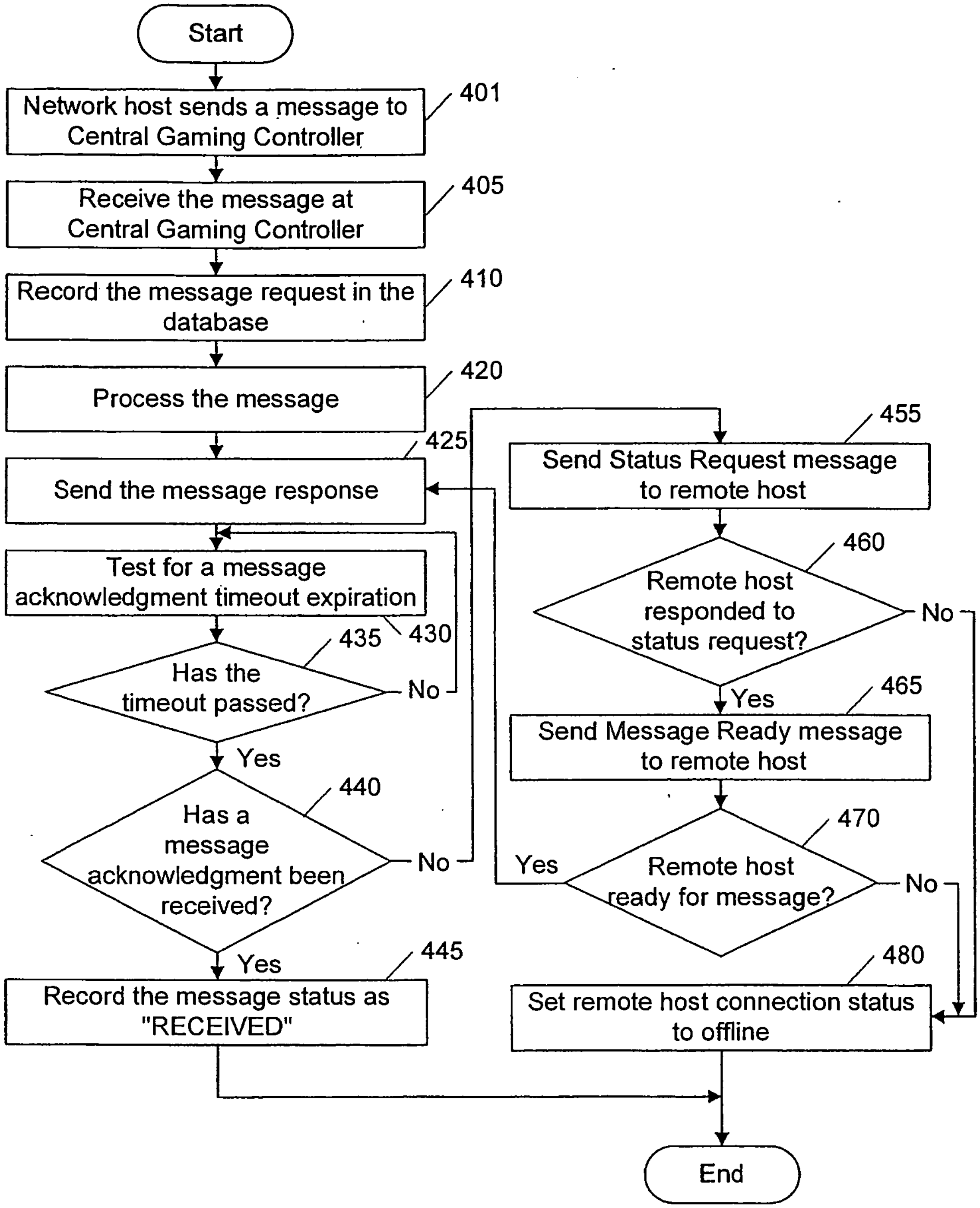

300
320
Controls
310
Display
395
Remote Control
110
Remote Player Device
Wireless Link 394
380
Smart Card Reader
390
Biometric Identity Device

FIG. 3



Start
Network host sends a message to Central Gaming Controller
401
Receive the message at Central Gaming Controller
405
Record the message request in the database
410
Process the message
420
Send the message response
425
Test for a message acknowledgment timeout expiration
430
Has the timeout passed?
435
No
Yes
Has a message acknowledgment been received?
440
No
Yes
Record the message status as "RECEIVED"
445
Send Status Request message to remote host
455
Remote host responded to status request?
460
No
Yes
Send Message Ready message to remote host
465
Remote host ready for message?
470
Yes
No
Set remote host connection status to offline
480
End

FIG. 4

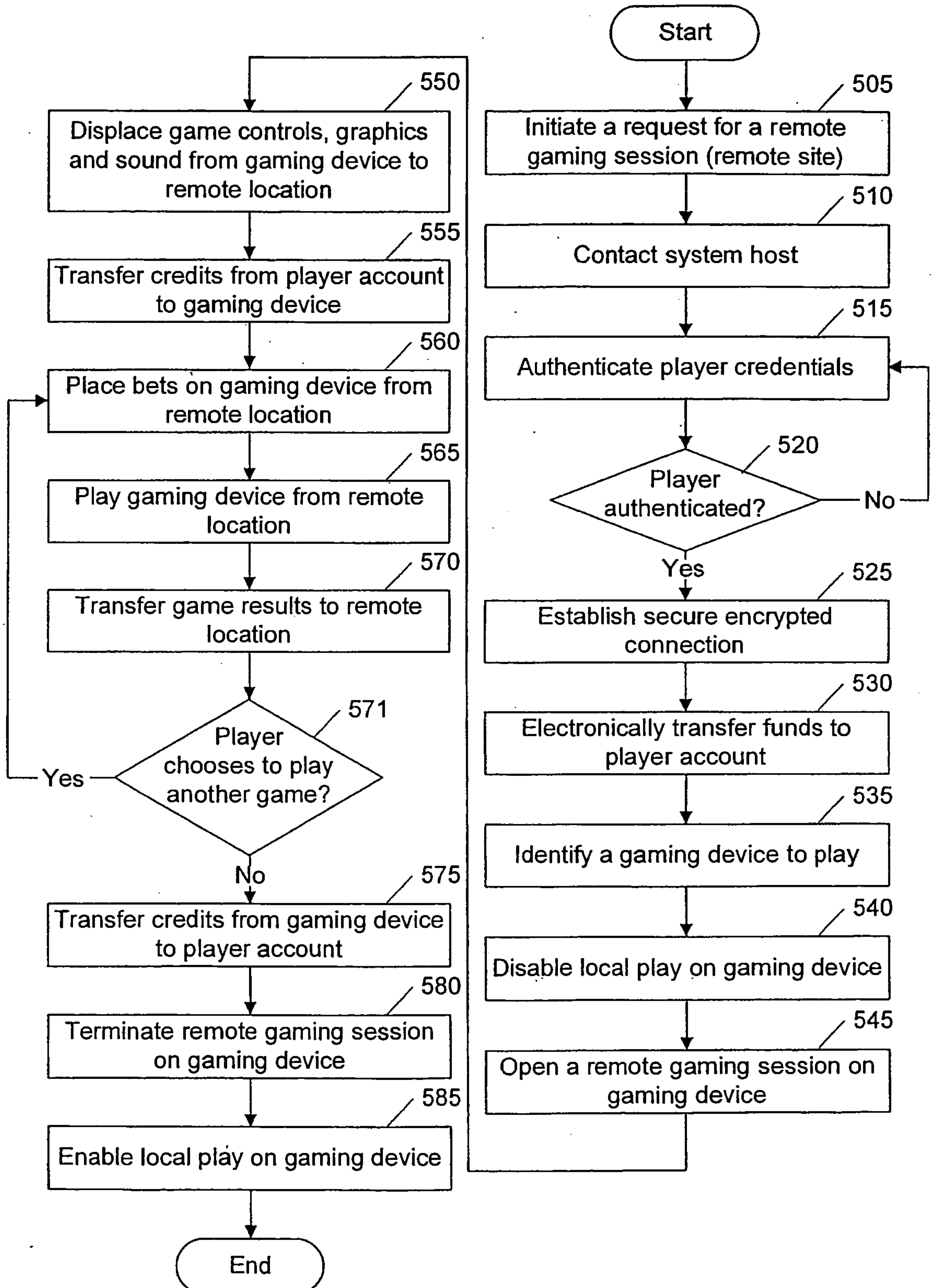

Start
505
Initiate a request for a remote gaming session (remote site)
510
Contact system host
515
Authenticate player credentials
520
Player authenticated?
No
Yes
525
Establish secure encrypted connection
530
Electronically transfer funds to player account
535
Identify a gaming device to play
540
Disable local play on gaming device
545
Open a remote gaming session on gaming device
550
Displace game controls, graphics and sound from gaming device to remote location
555
Transfer credits from player account to gaming device
560
Place bets on gaming device from remote location
565
Play gaming device from remote location
570
Transfer game results to remote location
571
Player chooses to play another game?
Yes
No
575
Transfer credits from gaming device to player account
580
Terminate remote gaming session on gaming device
585
Enable local play on gaming device
End

FIG. 5

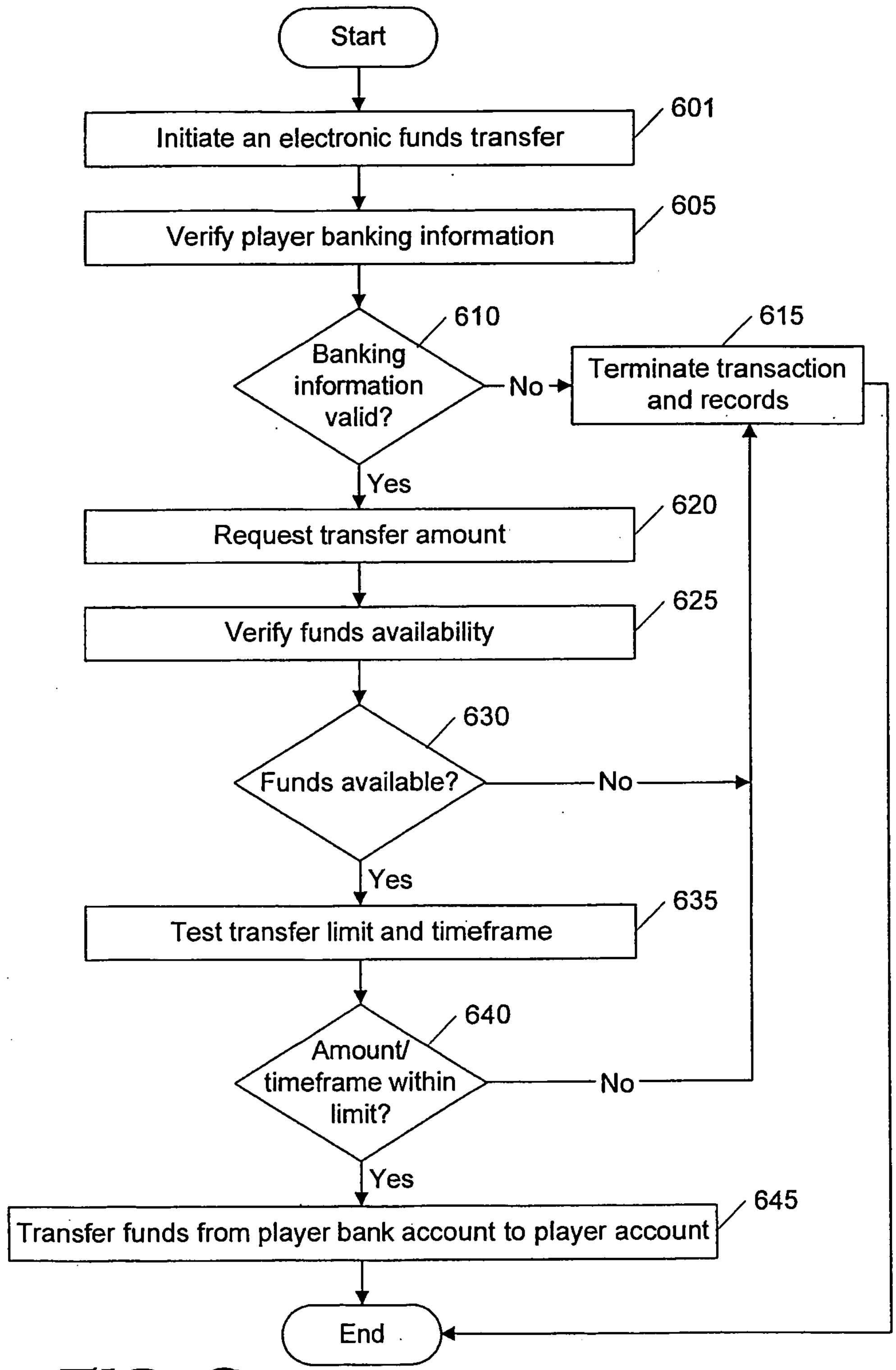

Start
Initiate an electronic funds transfer
601
Verify player banking information
605
Banking information valid?
610
No
Terminate transaction and records
615
Yes
Request transfer amount
620
Verify funds availability
625
Funds available?
630
No
Yes
Test transfer limit and timeframe
635
Amount/ timeframe within limit?
640
No
Yes
Transfer funds from player bank account to player account
645
End

FIG. 6

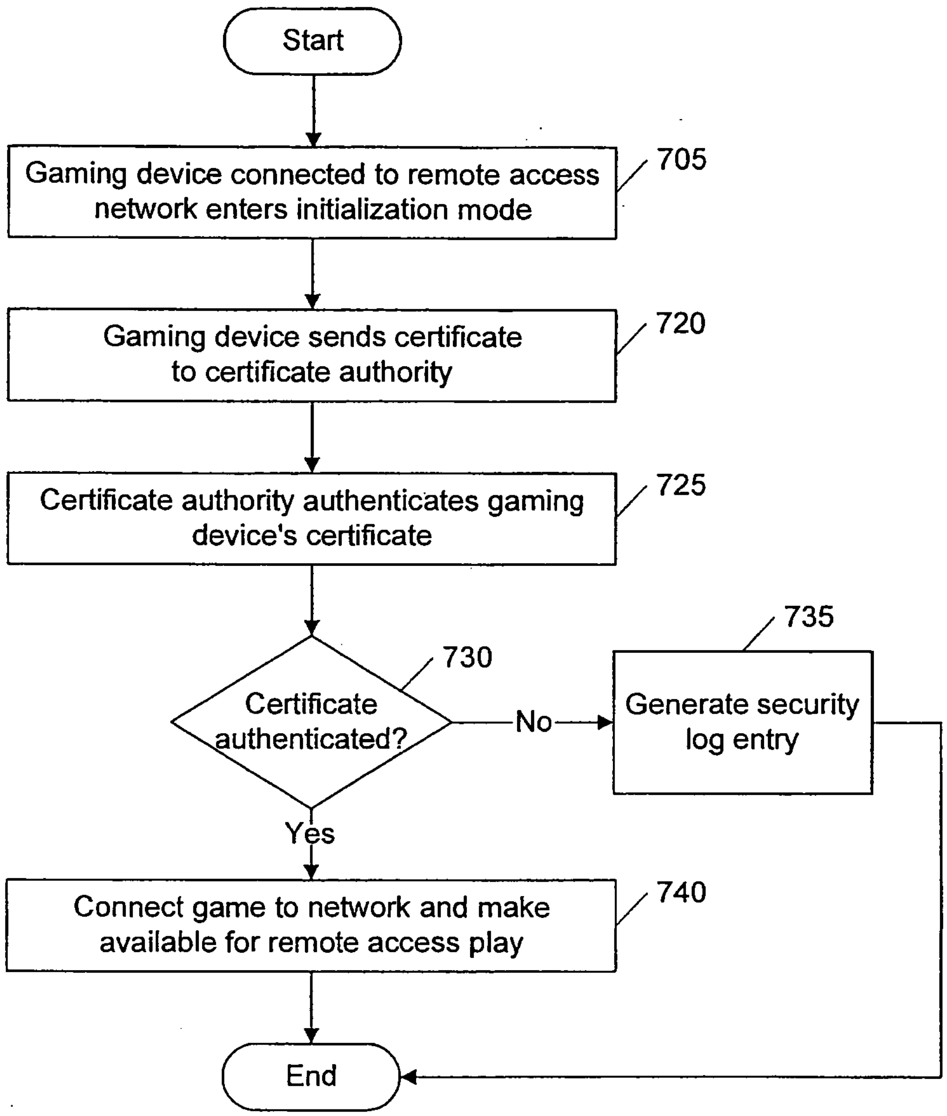

Start
Gaming device connected to remote access network enters initialization mode
705
Gaming device sends certificate to certificate authority
720
Certificate authority authenticates gaming device's certificate
725
Certificate authenticated?
730
No
Generate security log entry
735
Yes
Connect game to network and make available for remote access play
740
End

FIG. 7

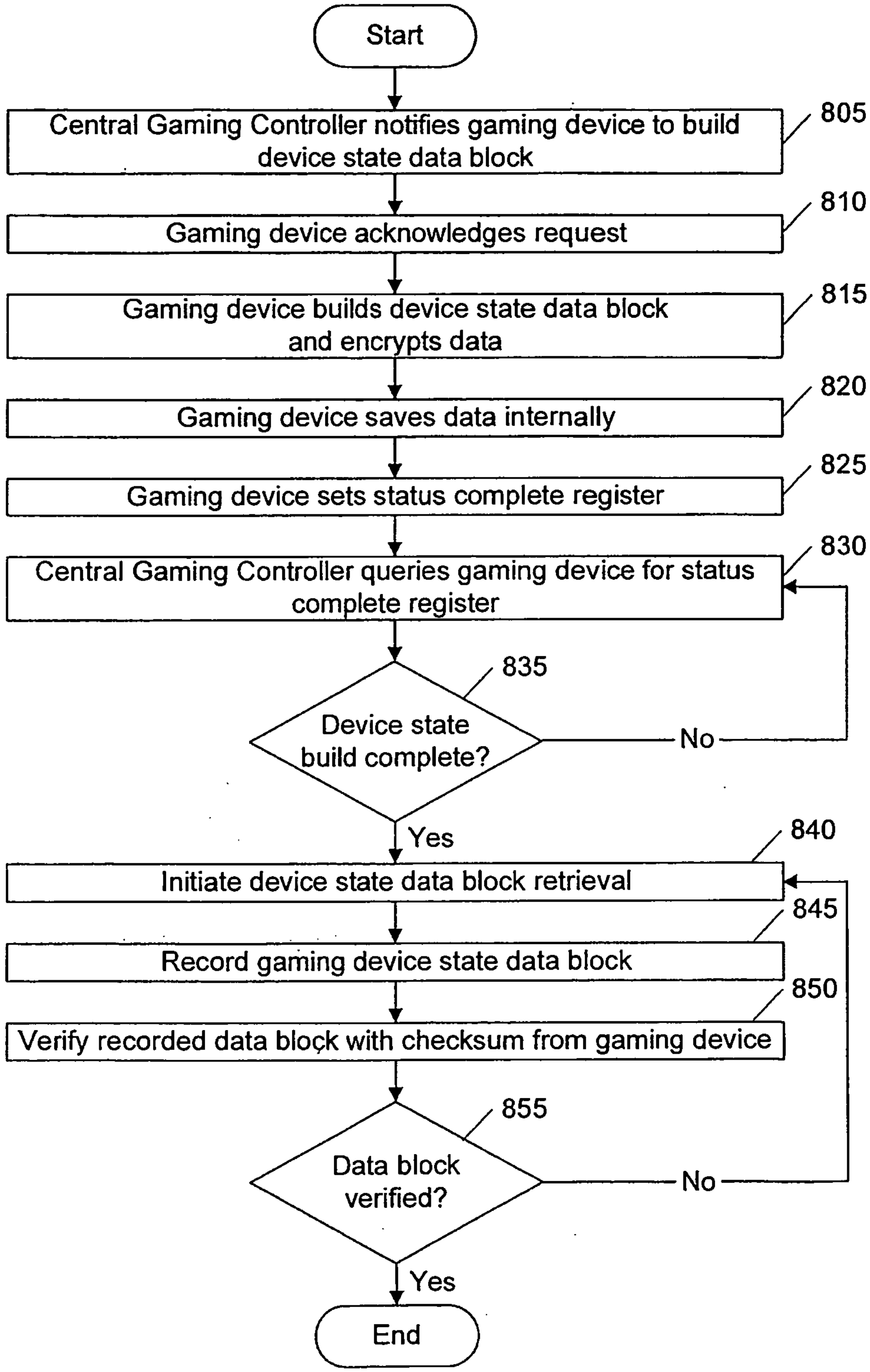

Start
Central Gaming Controller notifies gaming device to build device state data block
805
Gaming device acknowledges request
810
Gaming device builds device state data block and encrypts data
815
Gaming device saves data internally
820
Gaming device sets status complete register
825
Central Gaming Controller queries gaming device for status complete register
830
Device state build complete?
835
No
Yes
Initiate device state data block retrieval
840
Record gaming device state data block
845
Verify recorded data block with checksum from gaming device
850
Data block verified?
855
No
Yes
End

FIG. 8

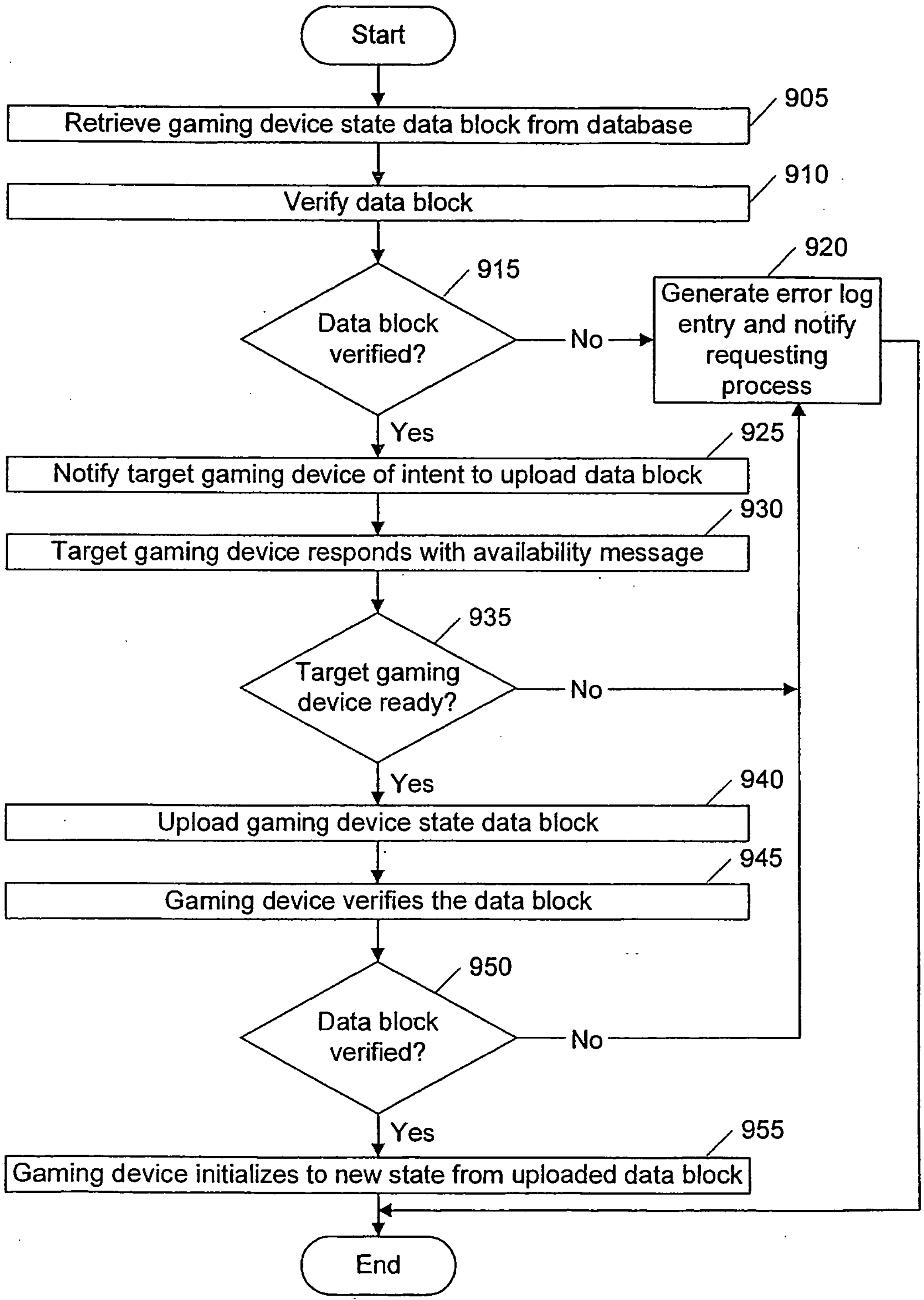

Start
905
Retrieve gaming device state data block from database
910
Verify data block
915
Data block verified?
No
920
Generate error log entry and notify requesting process
Yes
925
Notify target gaming device of intent to upload data block
930
Target gaming device responds with availability message
935
Target gaming device ready?
No
Yes
940
Upload gaming device state data block
945
Gaming device verifies the data block
950
Data block verified?
No
Yes
955
Gaming device initializes to new state from uploaded data block
End

FIG. 9

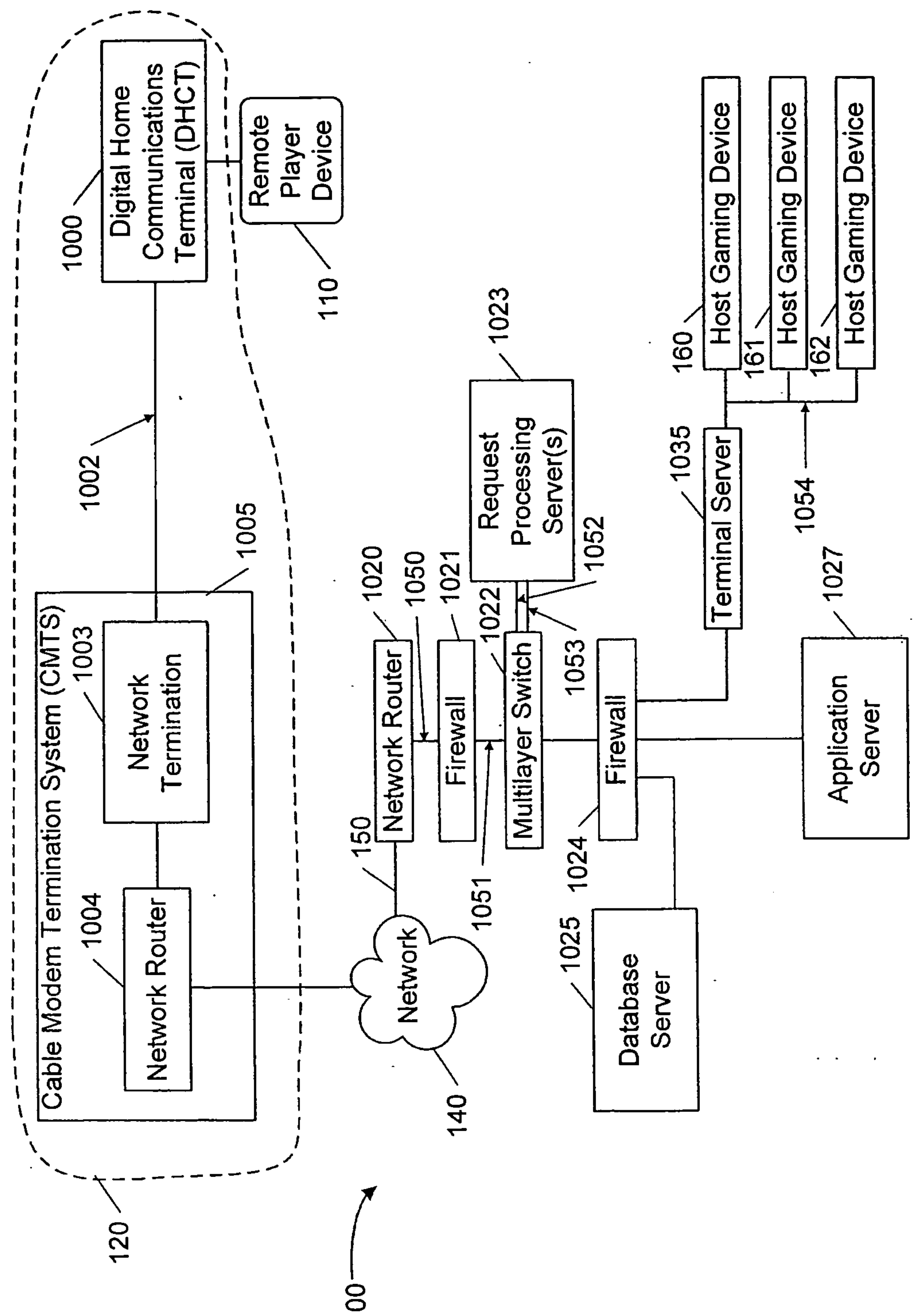
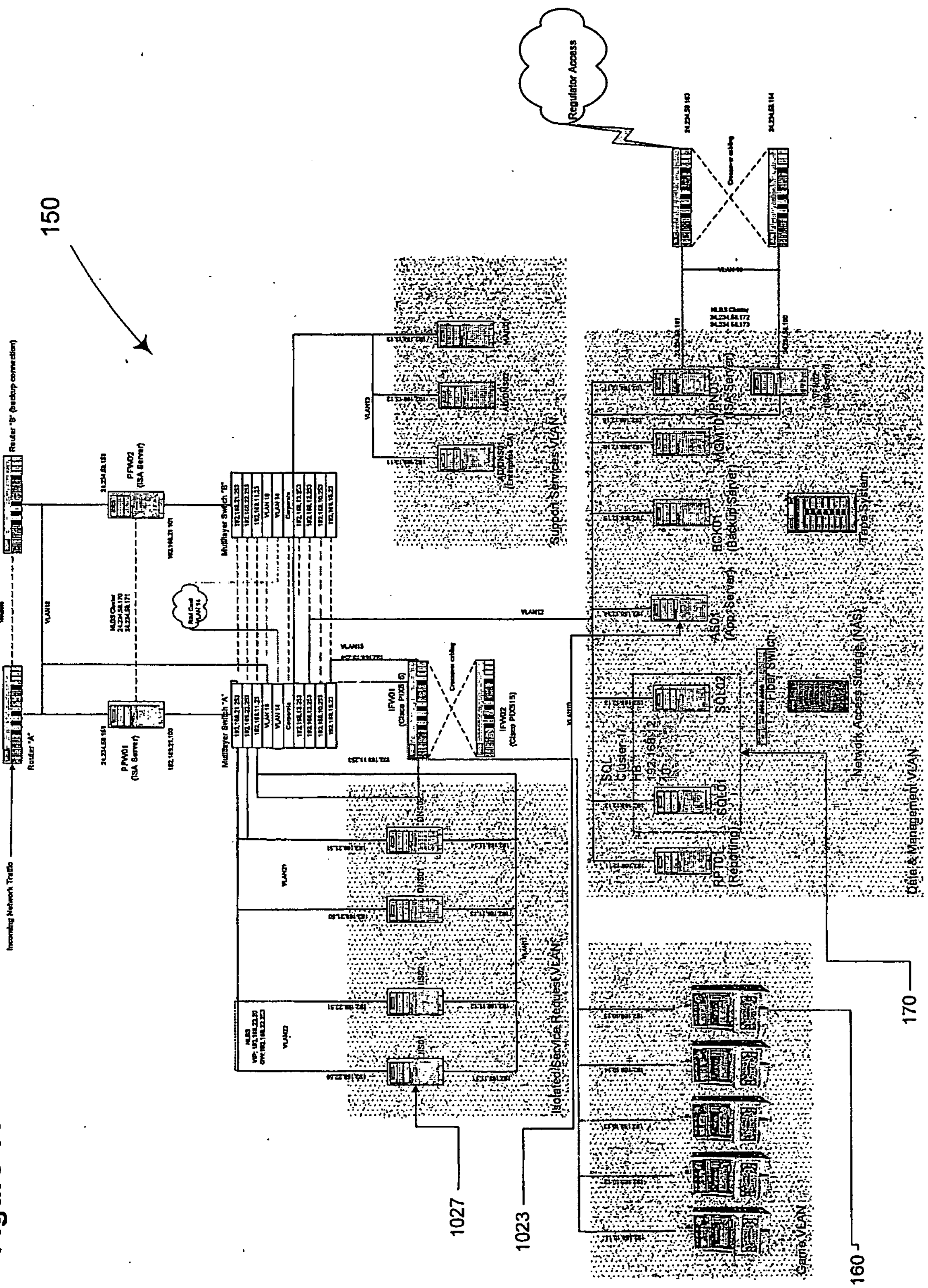

Digital Home Communications Terminal (DHCT)
1000
Remote Player Device
110
1002
Cable Modem Termination System (CMTS)
1005
Network Termination
1003
Network Router
1004
120
Network
140
150
Network Router
1020
1050
Firewall
1021
1022
Multilayer Switch
1051
Request Processing Server(s)
1023
1052
1053
Firewall
1024
Database Server
1025
Terminal Server
1035
Host Gaming Device
160
Host Gaming Device
161
Host Gaming Device
162
1054
Application Server
1027
100

FIG. 10



150
Regulator Access
Incoming Network Traffic
Router "A"
Router "B" (backup connection)
PFW01 (ISA Server)
PFW02 (ISA Server)
Multilayer Switch "A"
Multilayer Switch "B"
IFW01 (Cisco PIX515)
IFW02 (Cisco PIX515)
Support Services VLAN
Isolated Service Request VLAN
Data & Management VLAN
Network Access Storage (NAS)
Tape System
Game VLAN
1027
1023
160
170

Figure 11

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L

A2

(11) International Publication Number: WO 97/41661

(43) International Publication Date: 6 November 1997 (06.11.97)

(21) International Application Number: PCT/US97/06161

(22) International Filing Date: 16 April 1997 (16.04.97)

(30) Priority Data:
08/639,457 29 April 1996 (29.04.96) US

(71) Applicant: MOTOROLA INC. [US/US]; 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

(72) Inventor: DORENBOS, David; 241 N. Larch, Elmhurst, IL 60126 (US).

(74) Agents: LUKASIK, Susan, L. et al.; Motorola Inc., Intellectual Property Dept., 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

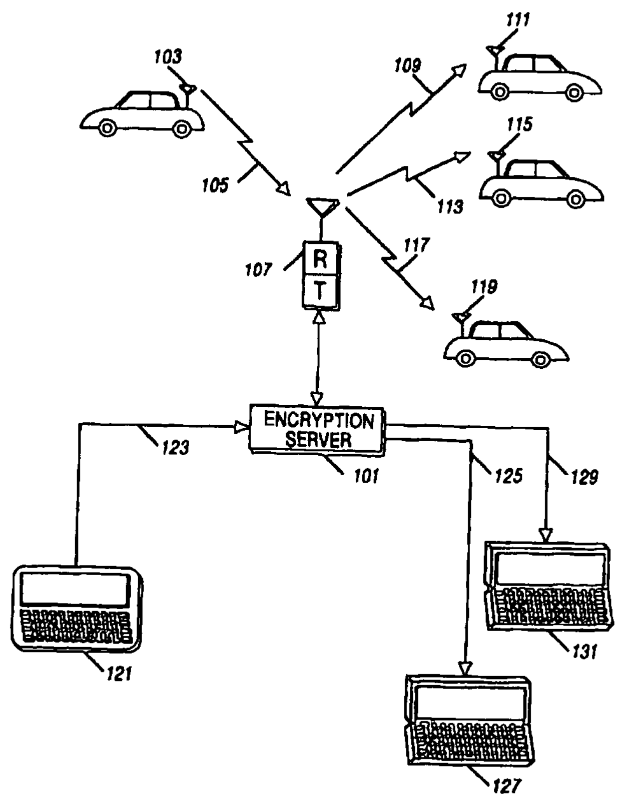
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG), European patent (AT, BE, CH, DE, DK, ES, FI, 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
Without international search report and to be republished upon receipt of that report.

(54) Title: USE OF AN ENCRYPTION SERVER FOR ENCRYPTING MESSAGES

(57) Abstract

An encryption server receives a first encrypted message (105) and decrypts (403) the encrypted message using a first key, yielding a decrypted message comprising a second encrypted message (105A), an identification of a sender of the first encrypted message, and an identification of a first recipient. The second encrypted message, the identification of the sender, and the identification of the first recipient are determined (405) from the decrypted message. The second encrypted message and the identification of the sender are encrypted (409) with a second key, yielding a third encrypted message (109). The third encrypted message (109) is transmitted to the first recipient.



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		

USE OF AN ENCRYPTION SERVER FOR ENCRYPTING MESSAGES

Field of the Invention

This invention relates to communication systems, including but not limited to encrypted communication systems.

Background of the Invention

Encrypted voice and data communication systems are well known. Many of these systems provide secure communications between two or more users by sharing one or more pieces of information between the users, which permits only those users knowing that information to properly decrypt the message. This information is known as the encryption key, or key for short. Encryption keys may be private keys, where a single key is utilized for encryption and decryption, or public keys, where multiple keys are utilized for encryption and decryption.

Methods of encrypting using public-key encryption are well known in the art. Typically, a public-key encryption is a method of encryption by which a single message is encrypted using a sender's private key and then a recipient's public key. The recipient then decrypts the message using the recipient's private key and then the sender's public key. Typically, public keys are 512 bits long, although some public keys have as few as 256 bits. Some encryption experts recommend using 1024-bit keys. Because the computational power required to break a key increases exponentially with the length of the key, longer keys provide more security. In addition, because two keys are needed to decrypt a message, two longer keys are more difficult to decrypt if neither key is known.

Today, secure communication systems are used to transmit data in an encrypted fashion. If a user wishes to send the same message to five different recipients, the user must encrypt the message five different times, each time using the public key of a different recipient for the message. The user then transmits the five messages to the five recipients. Such a process, however, is troublesome when the user wishes to transmit to, for example, 100 or more recipients. In this instance, the user must encrypt each message individually 100 or more times, one for each recipient. If the user has a portable communication device, such as a laptop computer, the user's battery may run out of power before encryption and transmission of each message has occurred. In addition, the encryption and transmission process can consume a lot of time and processing power for the portable device, rendering the portable device unavailable for other activities by the user during the encryption and transmission time period. Thus, such transmissions would be impractical for portable users.

Accordingly, there is a need for a method of transmitting encrypted data messages to multiple users without resulting in a time or power barrier to the user's communication device.

Brief Description of the Drawings

FIG. 1 is a block diagram of a communication system having an encryption server in accordance with the invention.

FIG. 2 is a block diagram of an encryption server in accordance with the invention.

FIG. 3 is a flowchart showing a method of transmission of a digital data message to an encryption server in accordance with the invention.

FIG. 4 is a flowchart showing a method of transmission of an encrypted message by an encryption server in accordance with the invention.

Description of a Preferred Embodiment

The following describes an apparatus for and method of using an encryption server for encrypting messages. Messages are encrypted twice, once with the sender's private key and then with an encryption server's public key before transmission of the messages to the encryption server. The encryption server decrypts received messages with the encryption server's private key, yielding an encrypted message, a user identification (ID), and one or more recipient IDs. The encryption server encrypts the encrypted message and the user ID individually with each of the recipient's public keys and transmits the resultant message(s) to the appropriate recipient. Each recipient decrypts the messages using the recipient's private key and the sender's public key. A secure communication system is thereby provided, wherein portable communication devices are neither tied up nor drained of power because the device's user wishes to send a single encrypted message to multiple recipients.

A method of using an encryption server for encrypting messages comprises the steps of, at a communication unit operated by a user generating a digital data message. The digital data message is encrypted using a first key, yielding a first encrypted message. An identification of the user and an identification of a first recipient are appended to the first encrypted message, yielding an appended first encrypted message. The appended first encrypted message is encrypted using a second key, yielding a second encrypted message. The second encrypted message is transmitted to an encryption server. At the encryption server, the second encrypted message is received. The second encrypted message is decrypted using a third key, yielding the appended first encrypted message. The first encrypted message, the identification of the user, and the identification of the first recipient are determined from the appended first encrypted message. The first encrypted message and the identification of the user are encrypted with a fourth key, yielding a third encrypted message. The third encrypted message is transmitted to the first recipient. In the preferred embodiment, the first key is a private key associated with the user, the second key is a public key associated with the encryption server, the third key is a private key associated with the encryption server, and the fourth key is a public key associated with the first recipient. Alternatively, the second key and the third key may be identical. The transmitting steps may be performed over wireless communication resources, such as radio frequency communication resources, or wireline communication resources, such as standard telephone lines or fiber optic cable.

In addition, the step of appending may further comprise the step of appending an identification of a second recipient to the first encrypted message, thereby yielding the appended first encrypted message. In this case, the method further comprises the steps of encrypting, by the encryption server, the first encrypted message and the identification of the user with a fifth key, yielding a fourth encrypted message, and transmitting the fourth encrypted message to the second recipient. In the preferred embodiment, the fifth key is a public key associated with the second recipient. Alternatively, the step of appending may comprise the step of appending three or more identifications of recipients to the first encrypted message, thereby yielding the appended first encrypted message.

A block diagram of a communication system having an encryption server is shown in FIG. 1. An encryption server 101 is shown at the center of FIG. 1. Further details of the encryption server 101 are shown in FIG. 2 described below. A user of a first communication unit 103 utilizes the first communication unit 103 to generate an digital data message that is encrypted in two stages in the preferred embodiment. In the first stage, the digital data message is encrypted using a first key, which is the user's private key in the preferred embodiment. The result of this encryption is a first-stage encrypted message. (In an alternate embodiment, the digital data message is not encrypted using the first key.) The user's identification (ID) and one or more recipient IDs are appended to the first-stage encrypted message, yielding an appended message. The appended message is encrypted using a second key, yielding a second-stage encrypted message 105. In the preferred embodiment, the second key is the public key associated with the encryption server 101. The communication unit transmits the second-stage encrypted message 105 to the encryption server via a wireless communication link to a wireless communication device 107, such as a radio frequency (RF) base station, repeater, or radio, or infrared communication device. The second-stage encrypted message 105 is conveyed by the wireless communication device 107 to the encryption server 101.

The encryption server 101 decrypts the second-stage encrypted message 105 using an appropriate key. In the preferred embodiment, the appropriate key is the encryption server's private key. The encryption server 101 then determines the user's ID from the decrypted message and also determines the IDs of all recipients that the user indicated as intended targets of the first-stage encrypted message. The encryption server 101 then encrypts the user's ID along with the first-stage encrypted message by encrypting with the public key of the first recipient. The resultant message 109 is transmitted to the first recipient, who utilizes communication unit 111. The encryption server then encrypts the first-stage encrypted message along with the user's ID by encrypting with the public key of the second recipient and transmitting the resultant encrypted message 113 to the second recipient, who utilizes communication unit 115. This process continues until the encryption server reaches the last recipient ID on the user's list, and encrypts the first-stage encrypted message along with the user's ID by encrypting with the public key of the last recipient and transmitting the resultant encrypted message 117 to the last recipient, who utilizes communication unit 119.

The encryption server 101 may also receive user requests for encryption from wireline communication devices 121 via wireline channels. As with the wireless transmission, the encryption server decrypts the received message 123 using the private key of the encryption server, then encrypts the resultant message individually for each different recipient using the appropriate recipient's individual public key. These recipients may be wireline devices 127 and 131, which receive the messages 125 and 129 via wireline communication channels.

The above examples describe RF to RF transmission and wireline to wireline transmission of encrypted messages. Nevertheless, the method of the present invention is equally successful if a wireline device 121 requests transmission to wireless communication units 111, 115, and 119. Similarly, a wireless communication unit 103 may request transmission from the encryption server 101 to wireline communication devices 127 and 131. In addition, the recipients may be a combination of both wireless and wireline communication units 111, 115, 119, 127, and 131, regardless of whether the sender uses a wireless communication unit 103 or a wireline communication device 121.

Upon receipt of the encrypted message from the encryption server, each recipient decrypts the message with the recipient's own private key, and after determining the user's ID, decrypts the resultant message with the user's public key, thereby yielding the original digital data message. The user is also referred to as the sender of the (second-stage) encrypted message 105.

A block diagram of an encryption server 101, including its input signals 105 and output signals 109, 113, 125, and 117, is shown in FIG. 2. In the preferred embodiment, the encryption server 101 is a Sun SparcServer2000 in a multiprocessor configuration, available from Sun Microsystems. The encryption server 101 comprises one or more processors 201, such as microprocessors or digital signal processors, as are well known in the art. The processors 201 have access to encryption and decryption algorithm(s) 203, a public key data base 205, and memory 211. The encryption/decryption algorithms 203 include public key algorithms, private algorithms, and other algorithms as may be used in the art. The public key data base 205 includes a list of IDs, as used by senders (users) and recipients, and the public keys associated with each of these IDs. The memory 211 includes programming and other data as is necessary to provide functionality as described herein for the encryption server 101. A receive block for wireline and wireless communications 207 and a transmit block for wireline and wireless communications 209 are also connected to the processors 201. The receive block for wireline and wireless communications 207 performs appropriate demodulation techniques on received messages 105 and 123. The transmit block for wireline and wireless communications 209 performs appropriate modulation techniques on messages 109, 113, 124, and 117 to be transmitted. In addition, the encryption server 101 may be equipped with hardware and/or software to provide the encryption server 101 with over-the-air-rekeying capabilities.

As shown in FIG. 2, a user message 105 comprises a second-stage encrypted (encrypted using the encryption server's public key) message comprising the digital data message 105A, first-stage encrypted with the user's (sender's) private key, in addition to the user ID and a number of recipient IDs. Alternatively, the user message 105 may comprise an unencrypted digital data message 105A, the user ID, and one or more recipient IDs. The user message 105 is input to the receive wireline/wireless block 207, the output of which is input to the processor(s) 201. The processor(s) 201 utilize(s) the encryption/decryption algorithm(s) 203 and the public key data base 205 to decrypt the message 105 using the private key of the encryption server. The processor(s) 201 then determine(s) the first-stage encrypted message 105A, the user ID, and the first recipient ID from the decrypted message. The processor(s) 201 then determine(s) the first recipient's public key from public key data base 205, and the encrypt the first-stage encrypted message 105A and the user ID by using the encryption/decryption algorithms 203 and the first recipient's public key. The processor(s) 201 then append(s) the first recipient ID, thereby yielding a message 109 that is sent to the transmit wireline/wireless block 209 for transmitting to the first recipient's communication unit 111, as shown in FIG. 1. A similar process is performed on the first-stage encrypted message (or unencrypted digital data message) 105A and the user ID for each of the recipients listed in the user's message 105.

In an alternate embodiment, the encryption server 101 may be physically distributed as one or more encryption servers. In this embodiment, the encryption server 101 encrypts the message using a second set of private and public keys associated with a second server. The message so encrypted is transmitted to the second encryption server. The second server decrypts the message and then encrypts the message using the public key(s) of the recipient(s). When traffic is heavy, the encryption server 101 may optimize its efficiency by determining the computation required to transmit directly to each recipient or transmit the request to one or more distributed servers. This process is transparent to the user.

The flowchart of FIG. 3 shows a method for use by a communication unit in transmitting a digital data message to an encryption server 101. At step 301, a digital data message is generated. If at step 303 the digital data message is not to be encrypted, the process continues with step 307. If at step 303 the digital data message is to be encrypted, the process continues with step 305, where the digital data message is encrypted using the private-key of the user who wishes to communicate the message. At step 307, it is determined if the IDs of the user and/or recipient(s) are to be encrypted. If the IDs are to be encrypted, the process continues with step 309, where the user ID and recipient ID(s) are appended to the encrypted message from step 305 or the unencrypted message from step 301 if no encryption took place. At step 311, the message from step 309, including the appended IDs, is encrypted using the public key of the encryption server 101. The process continues with step 317, where the encrypted message is transmitted to the encryption server 101. If at step 307 the IDs are not to be encrypted, the process continues with step 313, where the encrypted message of step 305 (or the unencrypted message from step 301 if no encryption took place) is encrypted with the public key of the encryption server 101. At step 315, the user ID and recipient ID(s) are appended to the encrypted message of step 313, and the process continues with step 317.

In an alternative embodiment, i.e., when the digital data message is not to be encrypted at step 303 of FIG. 3, the sender or user may decrypt the digital data message and, if desired, the recipient IDs only once, using the encryption server's public key. The encryption server then decrypts the message using the encryption server's private key, and encrypts the message individually for each of the recipients with the recipient's public key. The recipient then decrypts the message using only the recipient's private key. This method requires the user to locally store only one public key, the key of the encryption server. With this method, a single symmetrical key may be used to encrypt and decrypt the messages between the user and the encryption server 101, and one or more keys may be used to encrypt the messages between the encryption server and the recipient. Nevertheless, for better security, the encryption server 101 engaged in this embodiment should be a physically secured, e.g., locked away with limited access, because unencrypted information is present inside the encryption server 101. An advantage of such a system includes enabling law enforcement officials the ability to read the decrypted message as available in the encryption server 101.

The flowchart of FIG. 4 shows the method performed by the encryption server 101 in accordance with the present invention. At step 401, the encryption server receives the encrypted message transmitted by the communication unit 103. At step 403, the encryption server decrypts the message received at step 401 with the private key of the encryption server 101. At step 405, the encryption server determines the user ID, the recipient ID(s), and the encrypted (generated at step 305 of FIG. 3) or unencrypted (generated at step 301 of FIG. 3) data message. In an alternate embodiment, the encryption server 101 may be equipped with the appropriate keys to decrypt the digital data message 105A (when the message 105A is encrypted) so that law enforcement agencies may have full access to all information transmitted in the system.

At step 407, it is determined if the IDs (i.e., the user ID and/or recipient ID(s)) are to be encrypted before transmission. If the IDs are to be encrypted, the process continues with step 409, where the encryption server encrypts the encrypted data message along with the user ID, and the recipient's ID if desired, with the recipient's public key. At step 411, the encryption server transmits the encrypted message to the recipient whose public key was used at step 409. If at step 413 there are more recipients identified by the user to which the encryption server has not yet encrypted and transmitted the message, the process continues with step 407. If there are no more recipients at step 413, the process ends. If at step 407, the IDs are not to be encrypted, the process continues with step 415, where the encrypted data message is encrypted with the recipient's public key, and the user ID and the recipient's ID are appended to that encrypted message without further encryption, and the process continues with step 411.

Optionally, all messages may be encrypted at one time, and then transmitted in succession at one time, rather than encrypting a first message with one public key, then transmitting the encrypted first message right away, then encrypting a second message using another public key, and transmitting the encrypted second message immediately, and so forth.

The above text and associated drawings describe a method using public-key encryption. Private-key encryption, where the same key is used to encrypt and decrypt a message, may also be used. For example, the key used to encrypt the message send to the encryption server may be the same or identical key used to decrypt the encrypted message at the encryption server. In addition, the encryption method employed by the user to encrypt the original digital data message 105A may also be private-key encryption, rather than public-key encryption. In addition, a different encryption algorithm may be utilized for the user's first stage of encryption than for the user's second stage of encryption, the result of which is transmitted to the encryption server.

In the above manner, the encryption server encrypts the user's data message individually for each different recipient using that particular recipient's public key. The encryption server has more computing resources available to it than an individual communication unit, and can encrypt and transmit a message multiple times to many different users in a more efficient manner than can an individual communication unit. Individual communication units need not store all possible recipient's public keys, but instead need store only the encryption server's public key. Encryption of the recipient's ID(s) helps to secure the identity of the recipient(s) and eliminates a source of information for traffic analysis by undesired readers/interceptors of such information.

What is claimed is:

Claims

1. A method comprising the steps of:

at a communication unit operated by a user:

generating a digital data message;

encrypting the digital data message using a first key, yielding a first encrypted message;

appending an identification of the user and an identification of a first recipient to the first encrypted message, yielding an appended first encrypted message;

encrypting the appended first encrypted message using a second key, yielding a second encrypted message;

transmitting the second encrypted message to the encryption server, wherein the encryption server is not the first recipient.

2. The method of claim 1, wherein the first key is a private key associated with the user and wherein the second key is a public key associated with the encryption server.

3. The method of claim 1, wherein the step of appending further comprises the step of appending an identification of a second recipient to the first encrypted message, thereby yielding the appended first encrypted message.

4. A method comprising the steps of:

at an encryption server:

receiving a first encrypted message;

decrypting the encrypted message using a first key, yielding a decrypted message comprising a second encrypted message, an identification of a sender of the first encrypted message, and an identification of a first recipient;

determining the second encrypted message, the identification of the sender, and the identification of the first recipient from the decrypted message;

encrypting the second encrypted message and the identification of the sender with a second key, yielding a third encrypted message;

transmitting the third encrypted message to the first recipient.

5. The method of claim 4, wherein the first key is a private key associated with the encryption server and wherein the second key is a public key associated with the first recipient.

6. The method of claim 4, further comprising, when a second identification of a second recipient is part of the decrypted message, the steps of encrypting, by the encryption server, the second encrypted message and the identification of the sender with a third key, yielding a fourth encrypted message, and transmitting the fourth encrypted message to the second recipient.

7. A method comprising the steps of:

at a communication unit operated by a user:

generating a digital data message;

encrypting the digital data message using a first key, yielding a first encrypted message;

encrypting the first encrypted message using a second key, yielding a second encrypted message;

appending an identification of the user and an identification of a first recipient to the second encrypted message, yielding an appended second encrypted message;

transmitting the appended second encrypted message to the encryption server;

at the encryption server:

receiving the appended second encrypted message;

determining the second encrypted message, the identification of the user, and the identification of the first recipient from the appended second encrypted message;

decrypting the second encrypted message using a third key, yielding the first encrypted message;

encrypting the first encrypted message with a fourth key, yielding a third encrypted message;

transmitting the third encrypted message to the first recipient.

8. The method of claim 7, wherein the step of appending further comprises the step of appending an identification of a second recipient to the second encrypted message, thereby yielding the appended second encrypted message.

9. The method of claim 7, wherein the first key is a private key associated with the user, wherein the second key is a public key associated with the encryption server, wherein the third key is a private key associated with the encryption server, and wherein the fourth key is a public key associated with the first recipient.

10. The method of claim 7, wherein the identification of the user is encrypted using the second key before the step of appending.

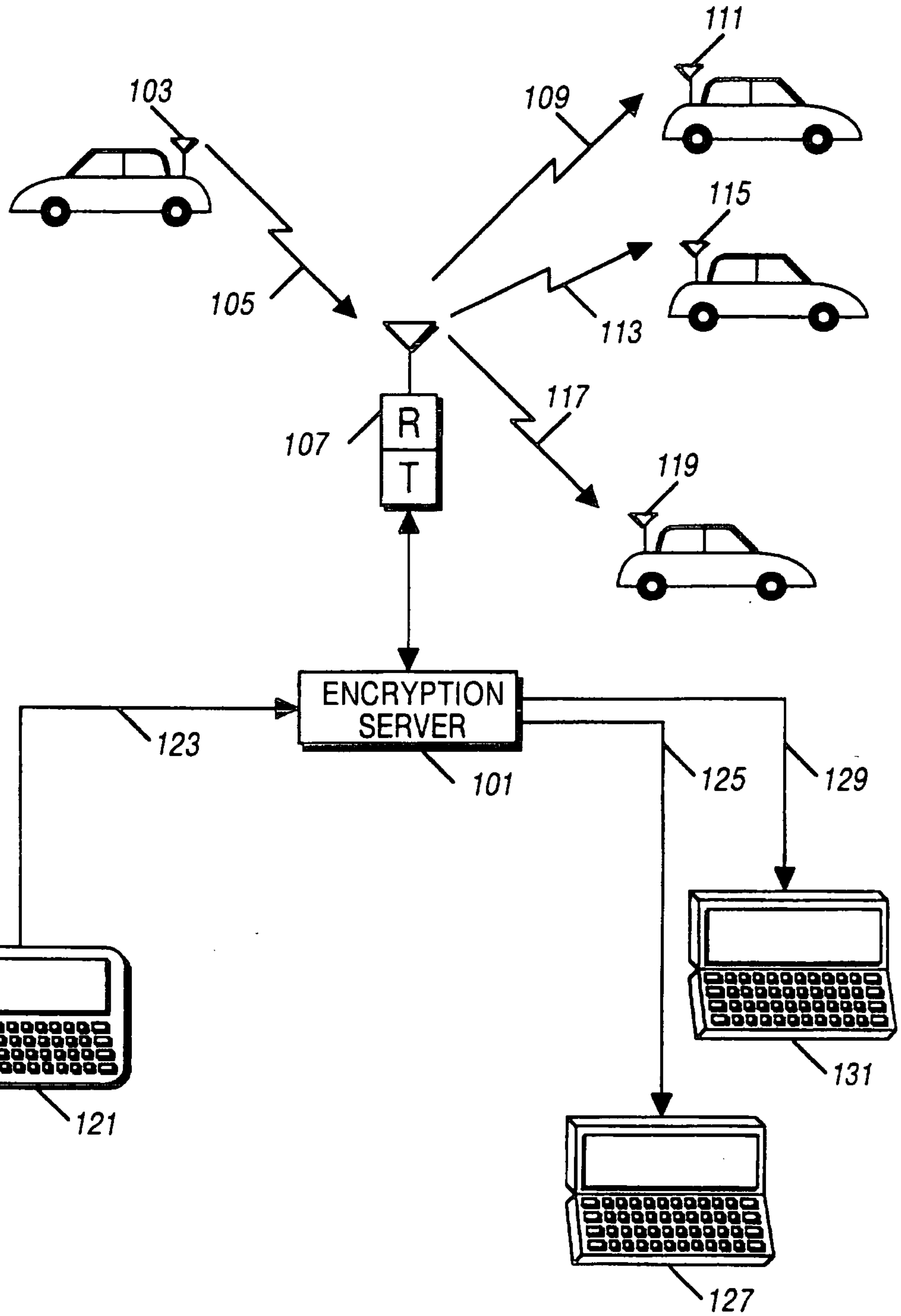

103
105
109
111
115
113
117
119
107
R
T
ENCRYPTION
SERVER
101
123
125
129
131
127
121

FIG. 1

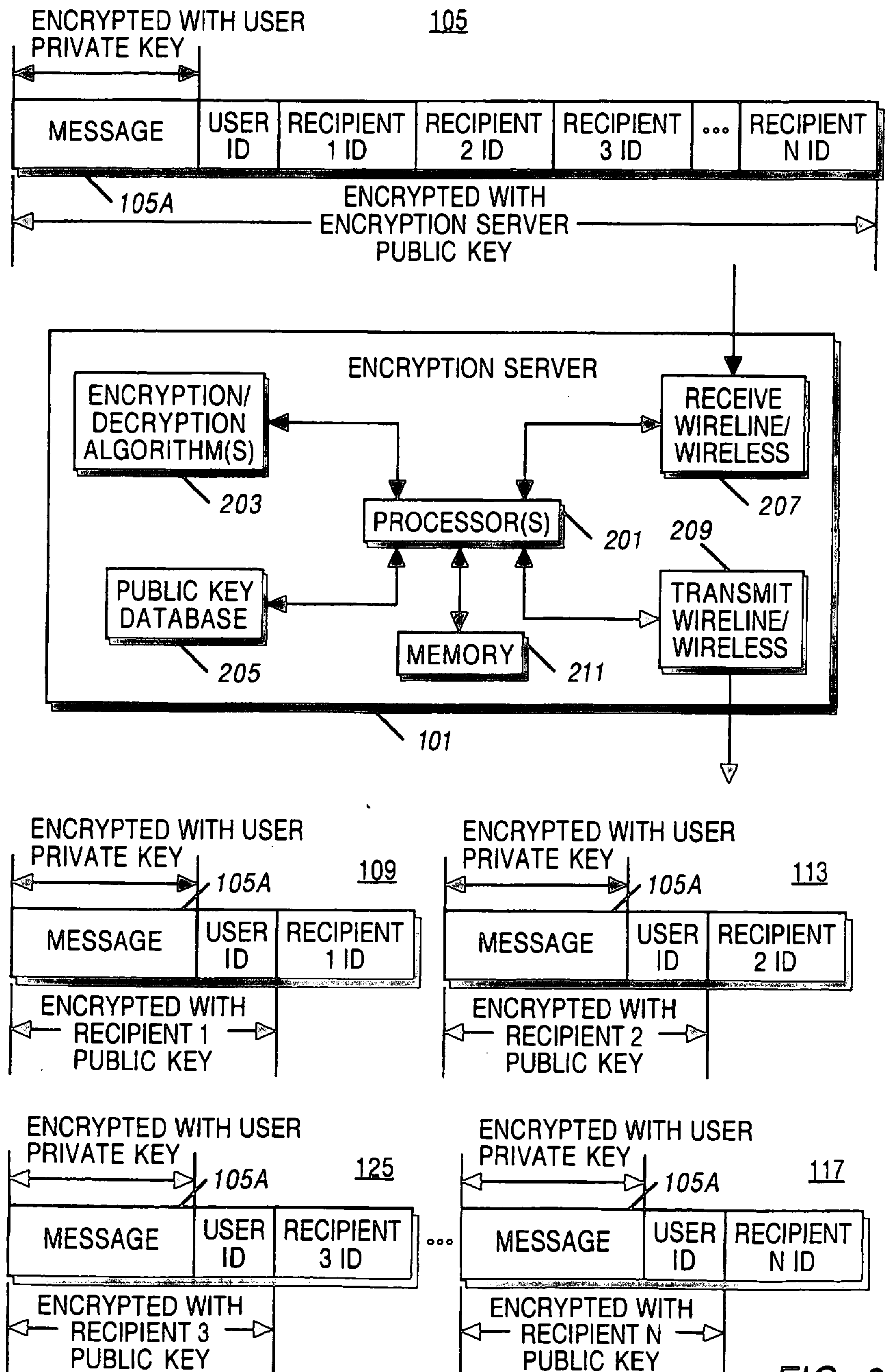

ENCRYPTED WITH USER PRIVATE KEY
105
MESSAGE
USER ID
RECIPIENT 1 ID
RECIPIENT 2 ID
RECIPIENT 3 ID
...
RECIPIENT N ID
105A
ENCRYPTED WITH ENCRYPTION SERVER PUBLIC KEY
ENCRYPTION SERVER
ENCRYPTION/ DECRYPTION ALGORITHM(S)
203
RECEIVE WIRELINE/ WIRELESS
207
PROCESSOR(S)
201
209
PUBLIC KEY DATABASE
205
MEMORY
211
TRANSMIT WIRELINE/ WIRELESS
101
ENCRYPTED WITH USER PRIVATE KEY
105A
109
MESSAGE
USER ID
RECIPIENT 1 ID
ENCRYPTED WITH RECIPIENT 1 PUBLIC KEY
ENCRYPTED WITH USER PRIVATE KEY
105A
113
MESSAGE
USER ID
RECIPIENT 2 ID
ENCRYPTED WITH RECIPIENT 2 PUBLIC KEY
ENCRYPTED WITH USER PRIVATE KEY
105A
125
MESSAGE
USER ID
RECIPIENT 3 ID
ENCRYPTED WITH RECIPIENT 3 PUBLIC KEY
...
ENCRYPTED WITH USER PRIVATE KEY
105A
117
MESSAGE
USER ID
RECIPIENT N ID
ENCRYPTED WITH RECIPIENT N PUBLIC KEY

FIG. 2

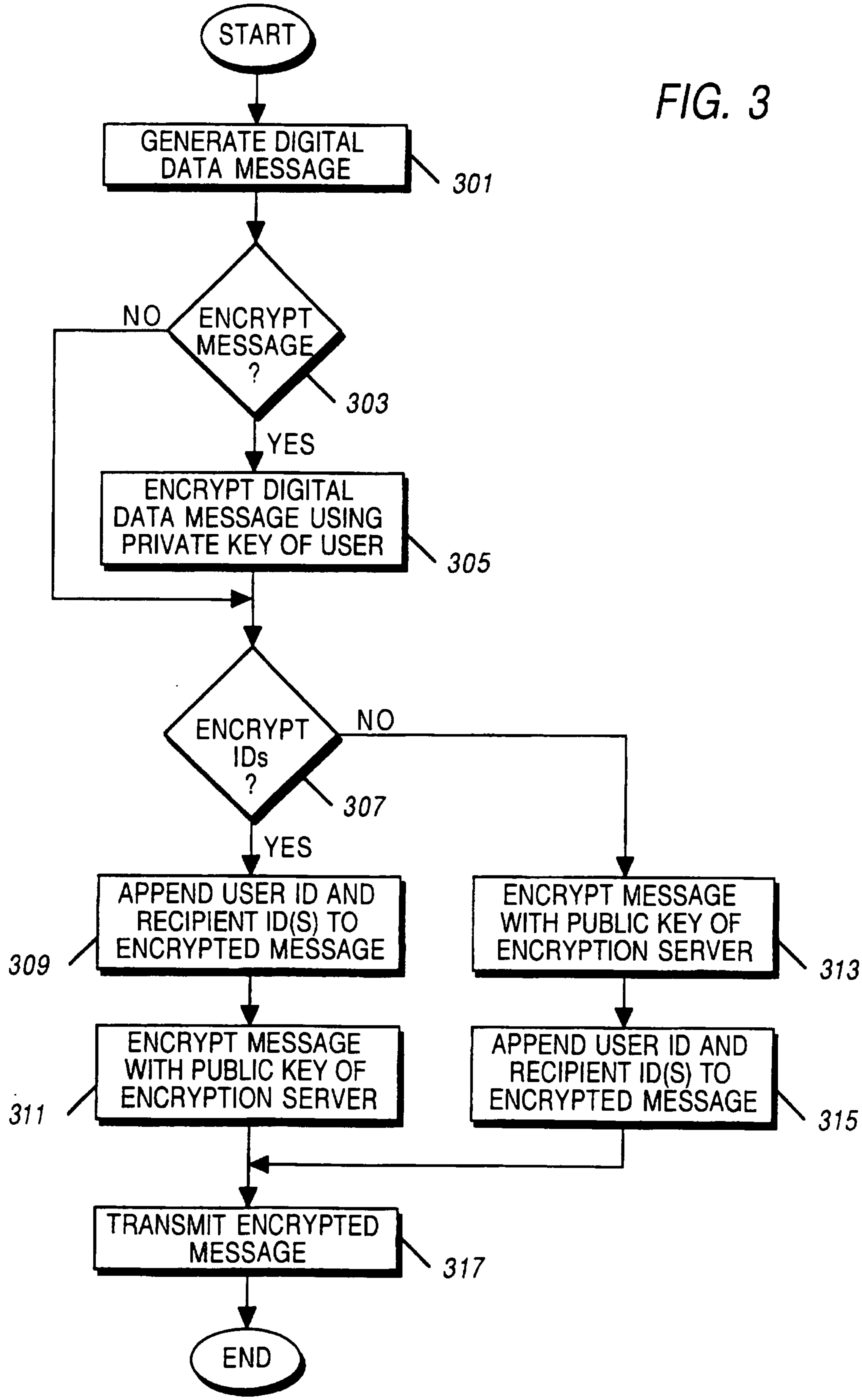
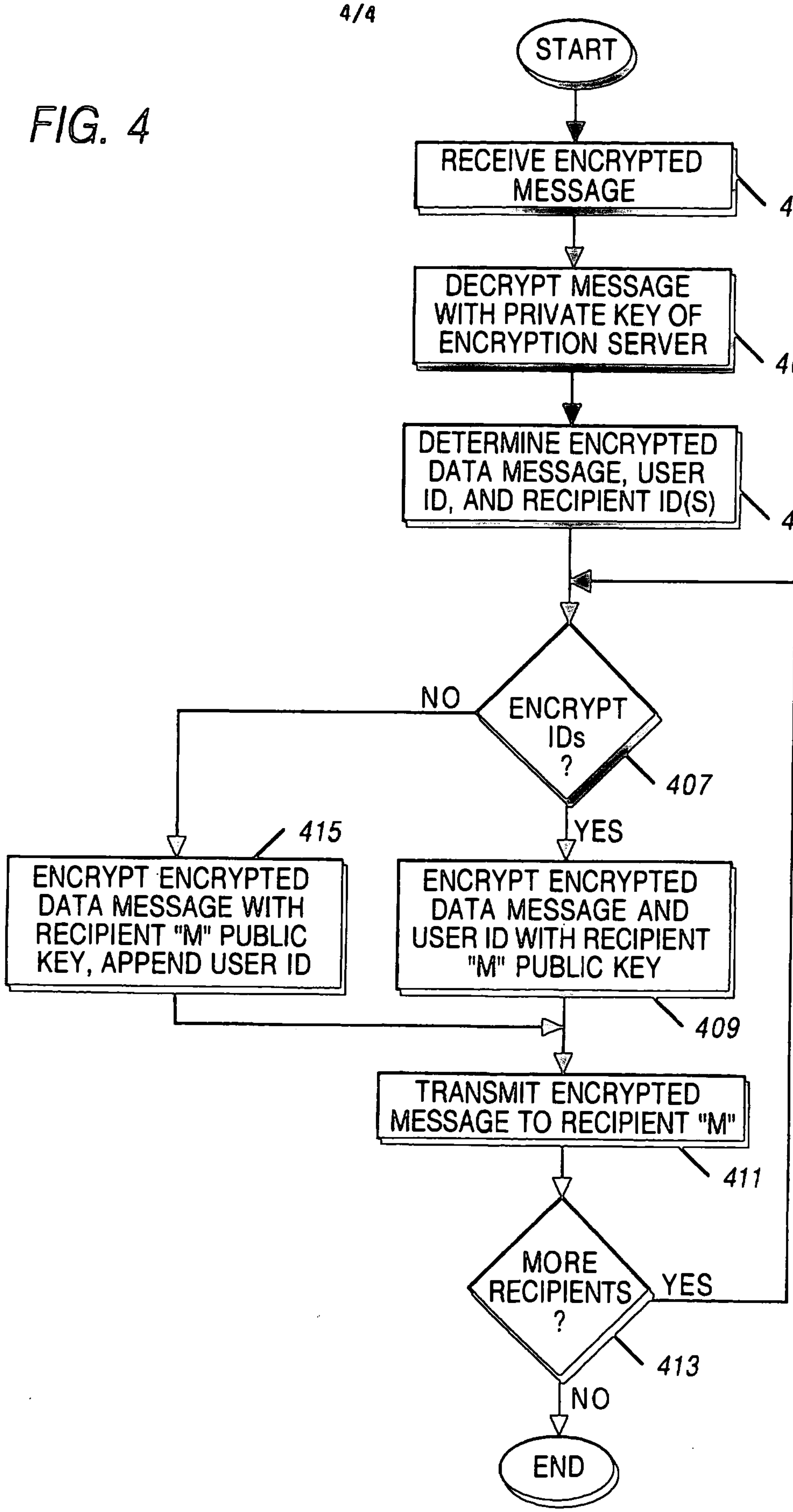

START
GENERATE DIGITAL DATA MESSAGE
301
ENCRYPT MESSAGE ?
NO
303
YES
ENCRYPT DIGITAL DATA MESSAGE USING PRIVATE KEY OF USER
305
ENCRYPT IDs ?
NO
307
YES
APPEND USER ID AND RECIPIENT ID(S) TO ENCRYPTED MESSAGE
309
ENCRYPT MESSAGE WITH PUBLIC KEY OF ENCRYPTION SERVER
313
ENCRYPT MESSAGE WITH PUBLIC KEY OF ENCRYPTION SERVER
311
APPEND USER ID AND RECIPIENT ID(S) TO ENCRYPTED MESSAGE
315
TRANSMIT ENCRYPTED MESSAGE
317
END

FIG. 3

FIG. 4


START
RECEIVE ENCRYPTED MESSAGE
401
DECRYPT MESSAGE WITH PRIVATE KEY OF ENCRYPTION SERVER
403
DETERMINE ENCRYPTED DATA MESSAGE, USER ID, AND RECIPIENT ID(S)
405
ENCRYPT IDs ?
407
NO
YES
ENCRYPT ENCRYPTED DATA MESSAGE WITH RECIPIENT "M" PUBLIC KEY, APPEND USER ID
415
ENCRYPT ENCRYPTED DATA MESSAGE AND USER ID WITH RECIPIENT "M" PUBLIC KEY
409
TRANSMIT ENCRYPTED MESSAGE TO RECIPIENT "M"
411
MORE RECIPIENTS ?
413
YES
NO
END

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04N 7/16, 7/167

A1

(11) International Publication Number: WO 98/43426

(43) International Publication Date: 1 October 1998 (01.10.98)

(21) International Application Number: PCT/EP97/02108

(22) International Filing Date: 25 April 1997 (25.04.97)

(30) Priority Data:
97400650.4 21 March 1997 (21.03.97) EP
(34) Countries for which the regional or international application was filed: FR et al.

(71) Applicant *(for all designated States except US)*: CANAL+ SOCIETE ANONYME [FR/FR]; 85/89, quai André Citroën, F-75711 Paris Cedex 15 (FR).

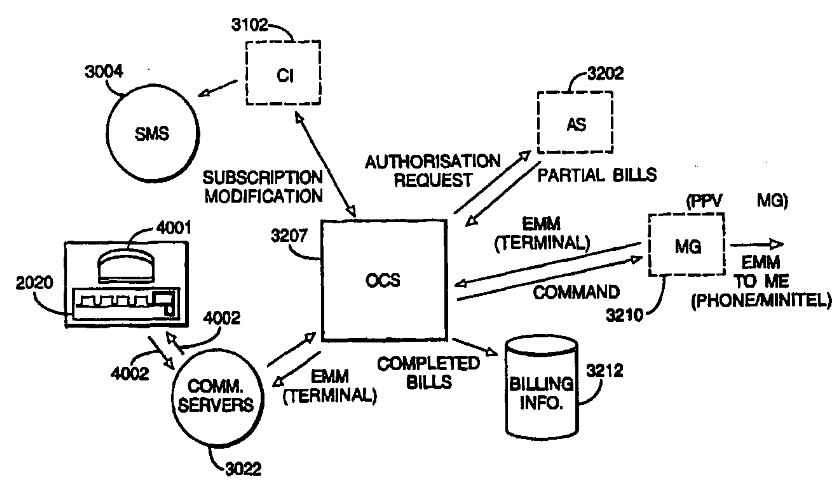
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: BAYASSI, Mulham [FR/FR]; 30, rue de Chambéry, F-75015 Paris (FR). DE LA TULLAYE, Pierre [FR/FR]; 7, allée Marcel Jouhandeau, F-92500 Rueil Malmaison (FR). JEZEQUEL, Jean-François [FR/FR]; 35, rue du Commandant Kieffer, F-95240 Cormeille en Parisis (FR).

(74) Agent: COZENS, Paul, Dennis; Mathys & Squire, 100 Grays Inn Road, London WC1X 8AL (GB).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, 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, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, 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
With international search report.

(54) Title: BROADCAST AND RECEPTION SYSTEM, AND CONDITIONAL ACCESS SYSTEM THEREFOR



(57) Abstract

A digital satellite television system has a plurality of set-top-boxes associated with a plurality of end users' television receivers, a modem and a decoder housed in each STB, a Subscriber Authorization System (SAS) incorporating or having associated therewith a plurality of communication servers, means included in the SAS for generating Electronic Managements Messages (EMM), a back channel interconnecting each of the STBs individually with the SAS, means included in the SAS and each STB so that the necessary information required to inject a relevant EMM into the system is supplied directly to the relevant communication server included in or associated with the SAS to authorise the release of the said EMM and/or means to connect the modem to the back channel and means whereby an EMM is transmissible to the decoder directly from a relevant communication server included in or associated with the SAS. Further important features are also disclosed.

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		

BROADCAST AND RECEPTION SYSTEM, AND CONDITIONAL ACCESS SYSTEM THEREFOR

The present invention relates to a broadcast and reception system, in particular to a mass-market digital interactive satellite television system, and to a conditional access system therefor.

In particular, but not exclusively, the invention relates to a mass-market broadcast system having some or all of the following preferred features:-

- o It is an information broadcast system, preferably a radio and/or television broadcast system
- o It is a satellite system (although it could be applicable to cable or terrestrial transmission)
- o It is a digital system, preferably using the MPEG, more preferably the MPEG-2, compression system for data/signal transmission
- o It affords the possibility of interactivity.

More particularly the present invention relates to so-called pay television (or radio) where a user/viewer selects a programme/film/game to be viewed for which payment is to be made, this being referred to as a pay-per-view (PPV) or in the case of data to be downloaded a so-called pay-per-file (PPF).

With such known PPV or PPF systems a significant amount of time is required to be spent by the user/viewer in order to carry out the actions necessary to actually access the product being selected.

For example, in one known system the sequence of steps which have to be carried out are as follows:-

I) The user telephones a so-called Subscriber Management System (SMS) which in this known system includes a number of human operators which answer the subscriber's call and to whom the subscriber communicates the necessary information concerning the selected product and concerning the financial status of the subscriber to a so-called Subscriber Authorization System (SAS) which has included in it or associated with it a plurality of communications servers.

ii) The operator at the SMS then has to check the financial status of the user before authorising the connection from the communications servers to the user's television set so that the product can be delivered and viewed by the user.

In another known system the human operator is replaced by an automatic voice server so that when the user telephones the SMS he/she hears a voice activated recording to which the user conveys the same information as i) above.

This second arrangement reduces the delay inherent in the first described arrangement which can be more easily overloaded when large numbers of users are wishing to order a product at the same time.

However, even with this second arrangement the user is involved in inputting significant information in the form of lengthy serial numbers which operation provides plenty of scope for error as well as being time consuming.

The third known arrangement involves the user making use of existing screen based systems such as MINITEL in France and PRESTEL in the United Kingdom, which systems replace the voice activated server referred to above in connection with the second arrangement. The MINITEL and PRESTEL systems themselves incorporate a modem at the consumer end.

In all these known arrangements the user is involved in the expenditure of significant time and effort in inputting all the information necessary to enable the system to in effect authorize the transmission of the chosen product to the user's television set.

In the case of a satellite television system there is a further delay involved in the user actually receiving the product selected.

In PPV and PPF systems the key element in controlling the user's access to products are so-called Entitlement Management Messages (EMM) which have to be injected into the system in order to give the user product access. More particularly the EMMs are the mechanism by which the encrypted data representative of a product is decrypted for a particular individual user.

In known satellite television systems the EMMs are transmitted to the user's televisions via the satellite link at regular intervals in the MPEG-2 data stream. Thus in the case of a particular user's EMM there can be a significant delay of perhaps several minutes before the user's next EMM transmission arrives at that user's television set.

This transmission delay is in addition to the delay referred to earlier which is inherent in the user having to manually input certain data into the system. The cumulative effect of these two delays is that it may take perhaps typically five minutes for a user to be able to gain access to the selected product.

The present invention is concerned with overcoming this problem.

In a first aspect, the present invention provides a conditional access system comprising:

means for generating a plurality of (preferably conditional access) messages; and

means for receiving the messages, said receiving means being adapted to communicate with said generating means via a communications server connected directly to said generating means.

Preferably, the message is an entitlement message for transmission (for example by broadcast) to the receiving means, said generating means being adapted to generate entitlement messages in response to data received from said receiving means.

The generating means may be arranged to transmit a message as a packet of digital data to said receiving means either via said communications server or via a satellite transponder.

The receiving means may be connectable to said communications server via a modem and telephone link.

In a related aspect, the present invention provides a conditional access system for affording conditional access to subscribers, comprising:

a subscriber management system;

a subscriber authorization system coupled to the subscriber management system; and

a communications server; said server being connected directly to the subscriber authorization system.

The system may further comprise a receiver/decoder for the subscriber, the receiver/decoder being connectable to said communications server, and hence to said subscriber authorization system, via a modem and telephone link.

In a second aspect, the present invention provides a broadcast and reception system including a conditional access system as described above.

In a third aspect, the present invention provides a broadcast and reception system comprising:

means for generating a plurality of entitlement messages relating to broadcast programs;

means for receiving said messages from said generating means; and

means for connecting the receiving means to the generating means to receive said messages, said connecting means being capable of effecting a dedicated connection between the rcceiving means and the generating means.

The dedicated connection would typically be a hard-wired connection and/or a modemmed connection, with the possibility of the connection been made via a cellular telephone system. In other words, the dedicated connection is capable of forming a channel of communication (from point to point). This is in contrast to broadcasting of information through the air or ambient medium. The connecting means would typically be a modem at the receiving means.

Hence, in a closely related aspect, the present invention provides a broadcast and reception system comprising:

means for generating a plurality of entitlement messages relating to broadcast programs;

means for receiving said messages from said generating means via a modem; and

means for connecting said modem to said generating means and said receiving means.

The above features can afford the advantage of providing the user the necessary viewing authorization (via the EMM) more quickly than has hitherto been possible, partly because, since the SAS typically uses a smaller amount of computer code than the SMS, the SAS can operate more efficiently (and in real time), partly because the SAS can itself, directly, generate the requisite EMM, and partly because the EMM can be passed to the user or subscriber via a dedicated (typically modemmed) link.

Preferably, the generating means is connected to said modem via a communications server which is preferably included in or associated with said generating means.

The receiving means may be further adapted to receive said entitlement messages via a satellite transponder.

The receiving means may be a receiver/decoder comprising means for receiving a compressed MPEG-type signal, means for decoding the received signal to provide a television signal and means for supplying the television signal to a television.

Preferably, the receiving means is adapted to communicate with said generating means via said modem and connecting means. The receiving means may comprise means for reading a smartcard insertable thereinto by an end user, the smartcard having stored therein data to initiate automatically the transmission of a message from said receiving means to said generating means upon insertion of the smartcard by the end user.

In addition, the system may further comprise a voice link to enable the end user of the broadcast and reception system to communicate with the generating means.

It will be understood from the above that the present invention provides two arrangements by which the time it takes for an end user to access a desired product is reduced. Preferably both arrangements are employed to achieve the maximum time saving but either arrangement can be used individually.

According to a further aspect of the present invention, there is provided a broadcast and reception system, comprising, at the broadcast end:

a broadcast system including means for broadcasting a callback request;

and at the reception end:

a receiver including means for calling back the broadcast system in response to the callback request.

By providing that the broadcast system can request the receiver to call it back, the possibility is afforded of the broadcast system obtaining information from the receiver about the state of the receiver.

Preferably, the means for calling back the broadcast system includes a modem connectable to a telephone system. By using a modemmed back channel, a simple way of putting the invention into effect can be provided.

Preferably also, the means for calling back the broadcast system is arranged to transfer to the broadcast system information concerning the receiver. This information might include the number of remaining tokens, the number of pre–booked sessions, and so on.

Preferably, the broadcast system includes means for storing the information, so that it can be processed at a later time, as desired.

Preferably, the broadcast means is arranged to broadcast a callback request which includes a command that the callback be made at a given time, and the means for calling back the broadcast system is arranged to respond to said command. By arranging for the callback to be later than the actual request, greater flexibility can be imparted to the system.

The broadcasting means may be arranged to broadcast as the callback request one or more Entitlement Messages for broadcast.

Preferably, the broadcast system includes means for generating a check message (such as a random number) and passing this to the receiver, the receiver includes means for encrypting the check message and passing this to the broadcast system, and the broadcast system further includes means for decrypting the check message received from the receiver and comparing this with the original check message. In this way it can be checked whether the receiver is genuine.

Any of the above features may be combined together in any appropriate combination. They may also be provided, as appropriate, in method aspects.

Preferred features of the present invention will now be described, purely by way of example, with reference to the accompanying drawings, in which:-

Figure 1 shows the overall architecture of a digital television system according to the preferred embodiment of the present invention;

Figure 2 shows the architecture of a conditional access system of the digital television system;

Figure 3 shows the structure of an Entitlement Management Message used in the conditional access system;

Figure 4 is a schematic diagram of the hardware of a Subscriber Authorisation System (SAS) according to a preferred embodiment of the present invention;

Figure 5 is a schematic diagram of the architecture of the SAS;

Figure 6 is a schematic diagram of a Subscriber Technical Management server forming part of the SAS;

Figure 7 is a flow diagram of the procedure for automatic renewal of subscriptions as implemented by the SAS;

Figure 8 is a schematic diagram of a group subscription bitmap used in the automatic renewal procedure;

Figure 9 shows the structure of an EMM used in the automatic renewal procedure;

Figure 10 shows in detail the structure of the EMM;

Figure 11 is a schematic diagram of an order centralized server when used to receive commands directly through communications servers;

Figure 12 illustrates diagrammatically a part of Figure 2 showing one embodiment of the present invention;

Figure 13 is a schematic diagram of the order centralized server when used to receive commands from the subscriber authorization system to request a callback;

Figure 14 is a schematic diagram of the communications servers;

Figure 15 shows the manner in which EMM emission cycle rate is varied according to the timing of a PPV event;

Figure 16 is a schematic diagram of a Message Emitter used to emit EMMs;

Figure 17 is a schematic diagram showing the manner of storage of EMMs within the Message Emitter;

Figure 18 is a schematic diagram of a smartcard;

Figure 19 is a schematic diagram of an arrangement of zones in the memory of the smartcard; and

Figure 20 is a schematic diagram of a PPV event description.

An overview of a digital television broadcast and reception system 1000 according to the present invention is shown in Figure 1. The invention includes a mostly conventional digital television system 2000 which uses the known MPEG–2 compression system to transmit compressed digital signals. In more detail, 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. 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 set 2022. The receiver/decoder 2020 decodes the compressed MPEG–2 signal into a television signal for the television set 2022.

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 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 2020. Using the decoder 2020 and smartcard, the end user may purchase events in either a subscription mode or a pay-per-view mode.

An interactive system 4000, also connected to the multiplexer 2004 and the receiver/decoder 2020 and again located partly in the broadcast centre and partly in the decoder, enables the end user to interact with various applications via a modemmed back channel 4002.

The conditional access system 3000 is now described in more detail.

With reference to Figure 2, in overview 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 linkage 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.

First encrypting units in the form of ciphering units 3008 utilising "mother" smartcards 3010 are connected to the SAS by linkage 3012. Second encrypting units again in the form of ciphering units 3014 utilising mother smartcards 3016 are connected to the multiplexer 2004 by linkage 3018. The receiver/decoder 2020 receives a "daughter" smartcard 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 smartcard on request.

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.

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 smartcard 3010 and 3016 respectively, for each electronic card, one (card 3016) for encrypting the ECMs and one (card 3010) for encrypting the EMMs.

The operation of the conditional access system 3000 of the digital television system will now be described in more detail with reference to the various components of the television system 2000 and the conditional access system 3000.

Multiplexer and Scrambler

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.

The scrambler generates a control word used in the scrambling process and included in the MPEG-2 stream in the multiplexer 2004. The control word 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. 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. 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.

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 one 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 and transmitted on to the multiplexer and scrambler 2004.

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 2014. In respect of each scrambled component of the service, a separate ECM is required.

Programme Transmission

The multiplexer 2004 receives electrical signals comprising 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 transmits the scrambled programmes, the encrypted EMMs and the encrypted ECMs as electric signals 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

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.

If the programme is not scrambled, that is, no ECM has been transmitted with the MPEG-2 stream, the receiver/decoder 2020 decompresses the data and transforms the signal into a video signal for transmission to television set 2022.

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" smartcard 3020 of the end user. This slots into a housing in the receiver/decoder 2020. The daughter smartcard 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 for onward transmission to television set 2022.

<u>Subscriber Management System (SMS)</u>

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.

Each SMS 3004 transmits messages to the SAS 3002 via respective linkage 3006 which imply modifications to or creations of Entitlement Management Messages (EMMs) to be transmitted to end users.

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).

As described later, the SAS 3002 sends messages (typically requesting information such as call-back information or billing information) to the SMS 3004, so that it will be apparent that communication between the two is two-way.

Entitlement Management Messages (EMMs)

The EMM is a message dedicated to an individual end user (subscriber), or a group of end users, only (in contrast with an ECM, which is dedicated to one scrambled programme only or a set of scrambled programmes if part of the same commercial offer). 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.

Various specific types of EMM are used in putting the present invention into practice. 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. 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. 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 Operator Identifier (OPI). Finally, a "unique" EMM is addressed to the unique identifier of the smartcard.

The structure of a typical EMM is now described with reference to Figure 3. Basically, the EMM, which is implemented as a series of digital data bits, comprises a header 3060, the EMM proper 3062, and a signature 3064. The header 3060 in turn comprises a type identifier 3066 to identify whether the type is individual, group, audience or some other type, a length identifier 3068 which gives the length of the EMM, an optional address 3070 for the EMM, an operator identifier 3072 and a key identifier 3074. The EMM proper 3062 of course varies greatly according to its type. Finally, the signature 3064, which is typically of 8 bytes long, provides a number of checks against corruption of the remaining data in the EMM.

Subscriber Authorization System (SAS)

The messages generated by the SMS 3004 are passed via linkage 3006 to the Subscriber Authorization System (SAS) 3002, which in turn generates messages acknowledging receipt of the messages generated by the SMS 3004 and passes these acknowledgements to the SMS 3004.

As shown in Figure 4, at the hardware level the SAS comprises in known fashion a mainframe computer 3050 (in the preferred embodiment a DEC machine) connected to one or more keyboards 3052 for data and command input, one or more Visual Display Units (VDUs) 3054 for display of output information and data storage means 3056. Some redundancy in hardware may be provided.

At the software level the SAS runs, in the preferred embodiment on a standard open VMS operating system, a suite of software whose architecture is now described in overview with reference to Figure 5; it will be understood that the software could alteratively be implemented in hardware.

In overview the SAS comprises a Subscription Chain area 3100 to give rights for subscription mode and to renew the rights automatically each month, a Pay Per View Chain area 3200 to give rights for PPV events, and an EMM Injector 3300 for passing EMMs created by the Subscription and PPV chain areas to the multiplexer and scrambler 2004, 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.

One function of the SAS 3002 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 3002, according to those rights and to information received from the SMS 3004, generates EMMs for the subscriber.

The Subscription Chain area 3100 comprises a Command Interface (CI) 3102, a Subscriber Technical Management (STM) server 3104, a Message Generator (MG) 3106, and the Ciphering Unit 3008.

The PPV Chain area 3200 comprises an Authorisation Server (AS) 3202, a relational database 3204 for storing relevant details of the end users, a local blacklist database 3205, Database Servers 3206 for the database, an Order Centralized Server (OCS) 3207, a Server for Programme Broadcaster (SPB) 3208, a Message Generator (MG) 3210 whose function is basically the same as that for the Subscription Chain area and is hence not described further in any detail, and the Ciphering Unit 3008.

The EMM Injector 3300 comprises a plurality of Message Emitters (MEs) 3302, 3304, 3306 and 3308 and Software Multiplexers (SMUXs) 3310 and 3312. In the preferred embodiment, there are two MEs, 3302 and 3304 for the Message Generator 3106, with the other two MEs 3306 and 3308 for the Message Generator 3210. MEs 3302 and 3306 are connected to the SMUX 3310 whilst MEs 3304 and 3308 are connected to the SMUX 3312.

Each of the three main components of the SAS (the Subscription Chain area, the PPV Chain area and the EMM Injector) are now considered in more detail.

Subscription Chain Area

Considering first the Subscription Chain area 3100, the Command Interface 3102 is primarily for despatching messages from the SMS 3004 to the STM server 3104, as well as to the OCS 3206, and from the OCS to the SMS. The Command Interface takes as input from the SMS either direct commands or batch files containing commands. It performs syntactic analysis on the messages coming from the STM server, and is able to emit accurate messages when an error occurs in a message (parameter out of range, missing parameter, and so on). It traces incoming commands in textual form in a trace file 3110 and also in binary form in a replay file 3112 in order to be able to replay a series of commands. Traces can be disabled and the size of files limited.

Detailed discussion of the STM server 3104 is now provided with particular reference to Figure 6. The STM server is effectively the main engine of the Subscription Chain area, and has the purpose of managing free rights, the creation of new subscribers and the renewal of existing subscribers. As shown in the figure, commands are passed on to the Message Generator 3106, albeit in a different format from that in which the commands are passed to the STM server. For each command, the STM server is arranged to send an acknowledgement message to the CI only when the relevant command has been successfully processed and sent to the MG.

The STM server includes a subscriber database 3120, in which all the relevant parameters of the subscribers are stored (smartcard number, commercial offers, state, group and position in the group, and so on). The database performs semantic checks of the commands sent by the CI 3102 against the content of the database, and updates the database when the commands are valid.

The STM server further manages a First In First Out (FIFO) buffer 3122 between the STM server and the MG, as well as a backup disk FIFO 3124. The purpose of the FIFOs is to average the flow of commands from the CI if the MG is not able to respond for a while for any reason. They can also ensure that in the case of a crash of the STM server or MG no command will be lost, since the STM server is arranged to empty (that is, send to the MG) its FIFOs when restarted. The FIFOs are implemented as files.

The STM server includes at its core an automatic renewal server 3126 which automatically generates renewals, and, if required by the operators, free rights. In this context, the generation of renewals may be thought of as including the generation of rights for the first time, although it will be understood that the generation of new rights is initiated at the SMS. As will become apparent, the two can be treated by roughly the same commands and EMMs.

Having the STM separate from the SAS, and the automatic renewal server within the SAS rather than (in known systems) in the SMS 3004, is a particularly important feature, since it can significantly reduce the number of commands which need to be passed from the SMS to the SAS (bearing in mind that the SMS and SAS may be in different locations and operated by different operators). In fact, the two main commands required from the SMS are merely commands that a new subscription should be started and that an existing subscription should be stopped (for example in the case of non-payment). By minimising command exchange between the SMS and SAS, the possibility of failure of command transfer in the linkage 3006 between the two is reduced; also, the design of the SMS does not need to take into account the features of the conditional access system 3000 generally.

Automatic renewal proceeds in the fashion indicated in the flow diagram of Figure 7. In order to reduce bandwidth, and given that a very high percentage of all renewals are standard, renewal proceeds in groups of subscribers; in the preferred embodiments there are 256 individual subscribers per group. The flow diagram begins with the start step 3130, and proceeds to step 3132 where a monthly activation of the renewal function is made (although of course it will be appreciated that other frequencies are also possible). With a monthly frequency, rights are given to the end user for the current month and all of the following month, at which point they expire if not renewed.

In step 3134 the subscriber database 3120 is accessed in respect of each group and each individual within that group to determine whether rights for the particular individual are to be renewed.

In step 3136, a group subscription bitmap is set up according to the contents of the subscriber database, as shown in Figure 8. The bitmap comprises a group identifier (in this case Group 1 – "G1") 3138 and 256 individual subscriber zones 3140. The individual bits in the bitmap are set to 1 or zero according to whether or not the particular subscriber is to have his rights renewed. A typical set of binary data is shown in the figure.

In step 3142 the appropriate commands, including the group subscription bitmap, are passed to the Message Generator 3106. In step 3143 the Message Generator sets an obsolescence date to indicate to the smartcard the date beyond which the particular subscription EMM is not valid; typically this date is set as the end of the next month.

In step 3144 the Message Generator generates from the commands appropriate group subscription EMMs and asks the Ciphering Unit 3008 to cipher the EMMs, the ciphered EMMs being then passed to the EMM Injector 3300, which, in step 3146, injects the EMMs into the MPEG–2 data stream.

Step 3148 indicates that the above described procedure is repeated for each and every group. The process is finally brought to an end at stop step 3150.

The flow diagram described above with reference to Figure 7 relates in fact specifically to the renewal of subscriptions. The STM also manages in a similar way free audience rights and new subscribers.

In the case of free audience rights, available for specific television programmes or groups of such programmes, these are made available by the STM issuing a command to the Message Generator to generate appropriate audience EMMs (for a whole audience) with an obsolescence date a given number of days (or weeks) hence. The MG computes the precise obsolescence date based on the STM command.

In the case of new subscribers, these are dealt with in two stages. Firstly, on purchase the smartcard in the receiver/decoder 2020 (if desired by the operator) affords the subscriber free rights for a given period (typically a few days). This is achieved by generating a bitmap for the subscriber which includes the relevant obsolescence date.

The subscriber then passes his completed paperwork to the operator managing the subscriber (at the SMS). Once the paperwork has been processed, the SMS supplies to the SAS a start command for that particular subscriber. On receipt by the SAS of the start command, the STM commands the MG to assign a unique address to the new subscriber (with a particular group number and position within the group) and to generate a special, so–called "commercial offer" subscription EMM (as opposed to the more usual "group" subscription EMM used for renewals) to provide the particular subscriber with rights until the end of the next month. From this point renewal of the subscriber can occur automatically as described above. By this two stage process it is possible to grant new subscribers rights until the SMS issues a stop command.

It is to be noted that the commercial offer subscription EMM is used for new subscribers and for reactivation of existing subscribers. The group subscription EMM is used for renewal and suspension purposes.

With reference to Figure 9, a typical subscription EMM proper (that is, ignoring the header and signature) generated by the above procedure comprises the following main portions, namely typically a 256 bit subscription (or subscribers' group) bitmap 3152, 128 bits of management ciphering keys 3154 for the ciphering of the EMM, 64 bits of each exploitation ciphering key 3156 to enable the smartcard 3020 to decipher a control word to provide access to broadcast programmes, and 16 bits of obsolescence date 3158 to indicate the date beyond which the smartcard will ignore the EMM. In fact in the preferred embodiment three exploitation keys are provided, one set for the present month, one set for the next month, and one for resume purposes in the event of system failure.

In more detail, the group subscription EMM proper has all of the above components, except the management ciphering keys 3154. The commercial offer subscription EMM proper (which is for an individual subscriber) includes instead of the full subscribers' group bitmap 3152 the group ID followed by the position in the group, and then management ciphering keys 3154 and three exploitation keys 3156, followed by the relevant obsolescence date 3158.

The Message Generator 3106 serves to transform commands issued by the STM server 3104 into EMMs for passing to the Message Emitter 3302. With reference to Figure 5, firstly, the MG produces the EMMs proper and passes them to the Ciphering Unit 3008 for ciphering with respect to the management and exploitation keys. The CU completes the signature 3064 on the EMM (see Figure 3) and passes the EMM back to the MG, where the header 3060 is added. The EMMs which are passed to the Message Emitter are thus complete EMMs. The Message Generator also 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 its cyclic transmission.

Again with reference to Figure 5, the Message Generator includes its own EMM database 3160 which, for the lifetime of the relevant EMM, stores it. It is erased once its emission duration has expired. The database is used to ensure consistency between the MG and ME, so that for example when an end user is suspended the ME will not continue to send renewals. In this regard the MG computes the relevant operations and sends them to the ME.

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.

Also concerning the Subscription Chain area, the Message Generator includes two FIFOs 3162 and 3164, one for each of the relevant Message Emitters 3302 and 3304 in the EMM Injector 3300, for storing the ciphered EMMs. Since the Subscription Chain area and EMM Injector may be a significant distance apart, the use of FIFOs can allow full continuity in EMM transmission even if the links 3166 and 3168 between the two fail. Similar FIFO's are provided in the Pay Per View Chain area.

One particular feature of the Message Generator in particular and the conditional access system in general concerns the way that it reduces the length of the EMM proper 3062 by mixing parameter length and identifier to save space. This is now described with reference to Figure 10 which illustrates an exemplary EMM (in fact a PPV EMM, which is the simplest EMM). The reduction in length occurs in the Pid (Packet or "Parameter" identifier) 3170. This comprises two portions, the actual ID 3172, and the length parameter for the packet 3174 (necessary in order that the start of the next packet can be identified). The whole Pid is expressed in just one byte of information, 4 bits being reserved for the ID, and four for the length. Because 4 bits is not sufficient to define the length in true binary fashion, a different correspondence between the bits and the actual length is used, this correspondence being represented in a look–up table, stored in storage area 3178 in the Message Generator (see Figure 5). The correspondence is typically as follows:–

0000	=	0
0001	=	1
0010	=	2
0011	=	3
0100	=	4
0101	=	5
0110	=	6
0111	=	7
1000	=	8
1001	=	9
1010	=	10
1011	=	11
1100	=	12
1101	=	16
1110	=	24
1111	=	32

It will be seen that the length parameter is not directly proportional to the actual length of the packet; the relationship is in part more quadratic rather than linear. This allows for a greater range of packet length.

Pay Per View Chain Area

Concerning the Pay Per View Chain area 3200, with reference to Figure 5 in more detail the Authorisation Server 3202 has as its client the Order Centralized Server 3207, which requests information about each subscriber which connects to the Communications Servers 3022 to purchase a PPV product.

If the subscriber is known from the AS 3202, a set of transactions takes place. If the subscriber is authorized for the order, the AS creates a bill and sends it to the OCS. Otherwise, it signals to the OCS that the order is not authorized.

It is only at the end of this set of transactions that the AS updates the end users database 3204 via the database servers (DBAS) 3206, if at least one transaction was authorized; this optimizes the number of database accesses.

The criteria according to which the AS authorizes purchase are stored in the database, accessed through DBAS processes. In one embodiment, the database is the same as the database accessed by the STM.

Depending on consumer profile, the authorization may be denied (PPV_Forbidden,Casino_Forbidden ...). These kind of criteria are updated by STM 3104, on behalf of the SMS 3004.

Other parameters are checked, such as limits allowed for purchase (either by credit card, automatic payment, or number of authorized token purchases per day).

In case of payment with a credit card, the number of the card is checked against a local blacklist stored in the local blacklist database 3205.

When all the verifications are successful, the AS:-

1. Generates a bill and sends it to the OCS, which completes this bill and stores it in a file, this file being later sent to the SMS for processing (customer actual billing); and

2. Updates the database, mainly to set new purchase limits.

This check-and-generate-bill-if-OK mechanism applies for each command a subscriber may request during a single connection (it is possible to order e.g. 5 movies in a single session).

It is to be noted that the AS has a reduced amount of information concerning the subscriber, by comparison with that held by the SMS. For example, the AS does not hold the name or address of the subscriber. On the other hand, the AS does hold the smartcard number of the subscriber, the subscriber's consumer category (so that different offers can be made to different subscribers), and various flags which state whether, for example, the subscriber may purchase on credit, or he is suspended or his smartcard has been stolen. Use of a reduced amount of information can help to reduce the amount of time taken to authorize a particular subscriber request.

The main purpose of the DBASs 3206 is to increase database performance seen from the AS, by paralleling the accesses (so actually it does not make much sense to define a configuration with only one DBAS). An AS parameter determines how many DBASes should connect. A given DBAS may be connected to only one AS.

The OCS 2307 mainly deals with PPV commands. It operates in several modes.

Firstly, it operates to process commands issued by the SMS, such as product refreshment (for instance, if the bill is already stored by the SMS, no bill is generated by the OCS), update of the wallet in the smartcard 3020, and session cancellation/update.

The various steps in the procedure are:-

1. Identifying the relevant subscriber (using the AS 3202);
2. If valid, generate adequate commands to the Message Generator, in order to send an appropriate EMM. Commands may be:

    Product commands,

    Update of the wallet,

Session erasure.

Note that these operations do not imply creation of billing information, since billing is already known from the SMS. These operations are assimilated to "free products" purchase.

Secondly, the OCS deals with commands received from the subscribers through the Communications Servers 3022. These may be received either via a modem connected to the receiver/decoder 2020, or by voice activation via the telephone 4001, or by key activation via a MINITEL, PRESTEL or like system where available.

Thirdly, the OCS deals with callback requests issued by the SMS. These last two modes of operation are now discussed in more detail.

In the second type of mode described above it was stated that the OCS deals with commands received directly from the end user (subscriber) through the Communications Servers 3022. These include product orders (such as for a particular PPV event), a subscription modification requested by the subscriber, and a reset of a parental code (a parental code being a code by which parents may restrict the right of access to certain programmes or classes of programmes).

The way in which these commands are dealt with is now described in more detail with reference to Figure 11.

Product orders by a subscriber involve the following steps:

1. Identifying through the AS the caller who is making a call through the CS 3022 ordering a particular product;
2. Checking the caller's request validity, again using the AS (where the order is placed using the receiver/decoder 2020, this is achieved by verifying the smartcard 3020 details);
3. Ascertain the price of the purchase;
4. Check that the price does not exceed the caller's credit limit etc;
5. Receiving a partial bill from the AS;

6. Filling additional fields in the bill to form a completed bill;
7. Adding the completed bill to a billing information storage file 3212 for later processing; and
8. Sending corresponding command(s) to the PPV Message Generator 3210 to generate the relevant EMM(s).

The EMM(s) is sent either on the modem line 4002 if the consumer placed the product order using the receiver/decoder 2020 (more details of this are described later), or else it is broadcast. The one exception to this is where there is some failure of the modem connection (in the case where the consumer places the order using the receiver/decoder); in this event the EMM is broadcast over the air.

A subscription modification requested by a subscriber involves:
1. Identifying the caller (using the AS);
2. Sending information to the Command Interface; the CI in turn forwards this information to the SMS; and
3. Via the CI, the OCS then receives an answer from the SMS (in terms of the cost of the modification, if the modification is possible).

If modification was requested using the receiver/decoder, the OCS generates a confirmation to the SMS. Otherwise, for example in the case of phone or Minitel, the subscriber is prompted for confirmation and this answer sent to the SMS via the OCS and the CI.

Reset of a parental code involves:
1. Identifying the caller (using AS); and
2. Sending a command to the MG to generate an appropriate EMM bearing an appropriate reset password.

In the case of reset of parental code, the command to reset the code is for security reasons not permitted to originate from the receiver/decoder. Only the SMS, telephone and MINITEL or like can originate such a command. Hence in this particular case the EMM(s) are broadcast only on air, never on the telephone line.

It will be understood from the above examples of different modes of operation of the OCS that the user can have direct access to the SAS, and in particular the OCS and AS, in that the Communications Servers are directly connected to the SAS, and in particular the OCS. This important feature is concerned with reducing the time for the user to communicate his command to the SAS.

This feature is illustrated further with reference to Figure 12, from which it can be seen that the end user's Set-Top-Box, and in particular its receiver/decoder 2020, has the capability of communicating directly with the Communications Servers 3022 associated with the SAS 3002. Instead of the connection from the end user to the Communications Servers 3022 of the SAS 3002 being through the SMS 3004 the connection is directly to the SAS 3002.

In fact, as directly mentioned two direct connections are provided.

The first direct connection is by a voice link via a telephone 4001 and appropriate telephone line (and/or by MINITEL or like connection where available) where the end users still have to input a series of voice commands or code numbers but time is saved compared with the communication being via the SMS 3004.

The second direct connection is from the receiver/decoder 2020 and the input of data is achieved automatically by the end user inserting his own daughter smartcard 3020 thus relieving the end user of the job of having to input the relevant data which in turn reduces the time taken and the likelihood of errors in making that input.

A further important feature which arises out of the above discussion is concerned with reducing the time taken for the resulting EMM to be transmitted to the end user in order to initiate viewing by the end user of the selected product.

In broad terms, and with reference to Figure 12, the feature is again achieved by providing the end user's receiver/decoder 2020 with the capability of communicating directly with the Communications Servers 3022 associated with the SAS 3002.

As described earlier the integrated receiver/decoder 2020 is connected directly to the Communications Servers 3022 by the modemmed back channel 4002 so that commands from the decoder 2020 are processed by the SAS 3002, messages generated (including EMMs) and then sent back directly to the decoder 2020 through the back channel 4002. A protocol is used in the communication between the CS 3022 and the receiver/decoder 2020 (as described later), so that the CS receive acknowledgement of receipt of the relevant EMM, thereby adding certainty to the procedure.

Thus, for example, in the case of a pre–book mode the SAS 3002 receives messages from the end user via the smartcard and decoder 2020 via its modem and via the telephone line 4002, requesting access to a specific event/product, and returns a suitable EMM via the telephone line 4002 and modem to the decoder 2020, the modem and decoder being preferably located together in a Set–Top–Box (STB). This is thus achieved without having to transmit the EMM in the MPEG–2 data stream 2002 via the multiplexer and scrambler 2004, the uplink 2012, satellite 2014 and datalink 2016 to enable the end user to view the event/product. This can save considerably on time and bandwidth. Virtual certainty is provided that as soon as the subscriber has paid for his purchase the EMM will arrive at the receiver/decoder 2020.

In the third type of mode of operation of the OCS 3207 described above, the OCS deals with callback requests issued by the SAS. This is illustrated with reference to Figure 13. Typical callback requests have the purpose of ensuring that the receiver/decoder 2020 calls back the SAS via the modemmed back channel 4002 with the information that the SAS requires of the receiver /decoder.

As instructed by the Command Interface 3102, the subscription chain Message Generator 3106 generates and sends to the receiver/decoder 202 a callback EMM. This EMM is ciphered by the Ciphering Unit 3008 for security reasons. The EMM may contain the time/date at which the receiver/decoder should wake up and perform a callback on its own, without being explicitly solicited; the EMM may also typically contain the phone numbers which the terminal must dial, the number of further attempts after unsuccessful calls and the delay between two calls.

When receiving the EMM, or at the specified time-date, the receiver/decoder connects to the Communications Servers 3022. The OCS 3207 first identifies the caller, using the AS 3202, and verifies certain details, such as smartcard operator and subscriber details. The OCS then asks the smartcard 3020 to send various ciphered information (such as the relevant session numbers, when the session was watched, how many times the subscriber is allowed to view the session again, the way in which the session was viewed, the number of remaining tokens, the number of prebooked sessions, etc). This information is deciphered by the PPV chain Message Generator 3210, again using the Ciphering Unit 3008. The OCS adds this information to a callback information storage file 3214 for later processing and passing to the SMS 3004. The information is ciphered for security reasons. The whole procedure is repeated until there is nothing more to be read from the smartcard.

One particular preferred feature of the callback facility is that before reading the smartcard (so just after the identification of the caller using the AS 3202 as described above) a check is made by the SAS 3002 that the receiver/decoder is indeed a genuine one rather than a pirated version or computer simulation. Such a check is carried out in the following manner. The SAS generates a random number, which is received by the receiver/decoder, ciphered, and then returned to the SAS. The SAS deciphers this number. If the deciphering is successful and the original random number is retrieved, it is concluded that the receiver/decoder is genuine, and the procedure continues. Otherwise, the procedure is discontinued.

Other functions which may occur during the callback are erasure of obsolete sessions on the smartcard, or filling of the wallet (this latter also being described later under the section entitled "Smartcard").

Also as regards the Pay Per View Chain area 3200, description is now made of the

Communications Servers 3022. At the hardware level, these comprise in the preferred embodiment a DEC Four parallel processor machine. At the software architecture level, with reference to Figure 14, in many respects the Communications Servers are conventional. One particular divergence from conventional designs arises from the fact that the Servers must serve both receiver/decoders 2020 and voice communication with conventional telephones 4001, as well possibly as MINITEL or like systems.

It will be noted in passing that two Order Centralized Servers 3207 are shown in Figure 14 (as "OCS1" and "OCS2"). Naturally any desired number may be provided.

The Communication Servers include two main servers ("CS1" and "CS2") as well as a number of frontal servers ("Frontal 1" and "Frontal 2"); whilst two frontal servers are shown in the figure, typically 10 or 12 may be provided per main server. Indeed, although two main servers CS1 and CS2 and two frontal servers, Frontal 1 and Frontal 2, have been shown, any number could be used. Some redundancy is usually desirable.

CS1 and CS2 are coupled to OCS1 and OCS2 via high level TCP/IP links 3230, whilst CS1 and CS2 are coupled to Frontal 1 and Frontal 2 via further TCP/IP links 3232.

As illustrated, CS1 and CS2 comprise servers for "SENDR" (transmission), "RECVR" (reception), "VTX" (MINITEL, PRESTEL or the like), "VOX" (voice communication), and "TRM" (communication with the receiver/decoder). These are coupled to the "BUS" for communication of signals to the Frontal servers.

CS1 and CS2 communicate directly with the receiver/decoders 2020 via their modemmed back channels 4002 using the X25 public network common protocol. The relatively low-level protocol between the Communications Servers 3022 and the receiver/decoders 3020 is in one preferred embodiment based upon the V42 standard international CCITT protocol, which provides reliability by having error detection and data re-transmission facilities, and uses a checksum routine to check the integrity of the re–transmission. An escape mechanism is also provided in order to prevent the transmission of disallowed characters.

On the other hand, voice telephone communication is carried out via the Frontal Communications Servers, each capable of picking up, say, 30 simultaneous voice connections from the connection 3234 to the local telephone network via the high speed "T2" (E1) standard telephony ISDN lines.

Three particular functions of the software portion of the Communications Servers (which could of course alternatively be implemented fully in hardware) are firstly to convert the relatively low level protocol information received from the receiver/decoder into the relatively high level protocol information output to the OCS, secondly to attenuate or control the number of simultaneous connections being made, and thirdly to provide several simultaneous channels without any mixing. In this last regard, the Communications Servers play the role of a form of multiplexer, with the interactions in a particular channel being defined by a given Session ID (identifier), which is in fact used throughout the communication chain.

Finally as regards the Pay Per View Chain area 3200, and with reference again to Figure 5, the Server for Programme Broadcast (SPB) 3208 is coupled to one or more Programme Broadcasters 3250 (which would typically be located remotely from the SAS) to receive programme information. The SPB filters out for further use information corresponding to PPV events (sessions).

A particularly important feature is that the filtered programme event information is passed by the SPB to the MG which in turn sends a directive (control command) to the ME to change the rate of cyclic emission of the EMMs in given circumstances; this is done by the ME finding all EMMs with the relevant session identifier and changing the cycle rate allocated to such EMMs. This feature might be thought of as a dynamic allocation of bandwidth for specific EMMs. Cyclic EMM emission is discussed in more detail in the section below concerned with the EMM Injector.

The circumstances in which the cycle rate is changed are now described with reference to Figure 15, which demonstrates how cycle rate 3252 is raised a short while (say 10 minutes) before a particular PPV programme event until the end of the event from a slow cycle rate of say once every 30 minutes to a fast cycle rate of say once every 30 seconds to 1 minute in order to meet the anticipated extra user demand for PPV events at those times. In this way bandwidth can be allocated dynamically according to the anticipated user demand. This can assist in reducing the overall bandwidth requirement.

The cycle rate of other EMMs may also be varied. For example the cycle rate of subscription EMMs may be varied by the Multiplexer and Scrambler 2004 sending the appropriate bitrate directive.

EMM Injector

Concerning the EMM Injector 3300, details of the Message Emitters 3302 to 3308, forming part of the EMM Injector and acting as output means for the Message Generator, are now described with reference to Figure 16. Their function is take the EMMs and to pass them cyclically (in the manner of a carousel) via respective links 3314 and 3316 to the Software Multiplexers 3310 and 3312 and thence to the hardware multiplexers and scramblers 2004. In return the software multiplexers and scramblers 2004 generate a global bitrate directive to control the overall cycling rate of the EMMs; to do so, the MEs take into account various parameters such as the cycle time, the size of EMM, and so on. In the figure, EMM_X and EMM _Y are group EMMs for operators X and Y, whilst EMM_Z are other EMMs for either operator X or operator Y.

Further description proceeds for an exemplary one of the Message Emitters; it will be appreciated that the remaining MEs operate in similar fashion. The ME operates under control of directives from the MG, most notably transmission start and stop time and emission rate, as well as session number if the EMM is a PPV EMM. In relation to the emission rate, in the preferred embodiment the relevant directive may take one of five values from Very fast to Very slow. The numeric values are not specified in the directive, but rather the ME maps the directive to an actual numeric value which is supplied by the relevant part of the SAS. In the preferred embodiment, the 5 emission rates are as follows:–

1.	Very fast	– every 30 seconds
2.	Fast	– every minute
3.	Medium	– every 15 minutes
4.	Slow	– every 30 minutes
5.	Very slow	– every 30 minutes

The ME has first and second databases 3320 and 3322. The first database is for those EMMs which have not yet achieved their broadcast date; these are stored in a series of chronological files in the database. The second database is for EMMs for immediate broadcast. In the event of a system crash, the ME is arranged to have the ability to re–read the relevant stored file and perform correct broadcast. All the files stored in the databases are updated upon request from the MG, when the MG wishes to maintain consistency between incoming directives and EMMs already sent to the ME. The EMMs actually being broadcast are also stored in Random Access Memory 3324.

A combination of the FIFOs 3162 and 3164 in the Message Generator and the databases 3320 and 3322 in the Message Emitter means that the two can operate in standalone mode if the link 3166 between them is temporarily broken; the ME can still broadcast EMMs.

The Software Multiplexers (SMUX) 3310 and 3312 provide an interface between the MEs and the hardware multiplexers 2004. In the preferred embodiment, they each receive EMMs from two of the MEs, although in general there is no restriction on the number of MEs that can be connected with one SMUX. The SMUXs concentrate the EMMs and then pass them according to the type of EMM to the appropriate hardware multiplexer. This is necessary because the hardware multiplexers take the different types of EMMs and place them at different places in the MPEG–2 stream. The

SMUX's also forward global bitrate directives from the hardware multiplexers to the MEs.

One particularly important feature of the ME is that it emits EMMs in random order. The reason for this is as follows. The Message Emitter has no ability to sense or control what it emits to the multiplexer. Hence it is possible that it may transmit two EMMs which are to be received and decoded by the receiver/decoder 2020 back to back. In such circumstances, further, it is possible that if the EMMs are insufficiently separated the receiver/decoder and smartcard will be unable to sense and decode properly the second of the EMMs. Cyclically emitting the EMMs in random order can solve this problem.

The manner in which randomization is achieved is now described with reference to Figure 17; in the preferred embodiment the necessary software logic is implemented in the ADA computer language. A particularly important part of the randomization is the correct storage of the EMMs in the databases 3320 and 3322 (which are used for backup purposes) and in the RAM 3324. For a particular cycle rate and operator, the EMMs are stored in a two–dimensional array, by rank 3330 (going say from A to Z) and number in the rank 3332 (going from 0 to N). A third dimension is added by cycle rate 3334, so that there are as many two–dimensional arrays as there are cycle rates. In the preferred embodiment there are 256 ranks and typically 200 or 300 EMMs in each rank; there are 5 cycle rates. A final dimension to the array is added by the presence of different operators; there are as many three–dimensional arrays as there are operators. Storage of the data in this fashion can permit rapid retrieval in the event that the MG wants to delete a particular EMM.

Storage of the EMMs takes place according to the "hash" algorithm (otherwise known as the "one–way hash function". This operates on a modulo approach, so that successive ranks are filled before a higher number in the rank is used, and the number of EMMs in each rank remains roughly constant. The example is considered of there being 256 ranks. When the MG sends the ME an EMM with identifer (ID) 1, the rank "1" is assigned to this EMM, and it takes the first number 3332 in the rank 3330.

The EMM with ID 2 is assigned the rank "2", and so on, up to the rank 256. The EMM with ID 257 is assigned the rank "1" again (based on the modulo function), and takes the second number in the first rank, and so on.

Retrieval of a specific EMM, for example when deletion of a specific EMM is requested by the MG, is effected by means of the inverse of the above. The hash algorithm is applied to the EMM ID to obtain the rank, after which the number in the rank is found.

The actual randomization occurs when the EMMs are, on a cyclical basis, retrieved from RAM 3324 using the randomization means 3340 which is implemented in the hardware and/or software of the Message Emitter. The retrieval is random, and again based on the hash algorithm. Firstly, a random number (in the above example initially in the range 1 to 256) is chosen, to yield the particular rank of interest. Secondly, a further random number is chosen to yield the particular number in the rank. The further random number is selected according to the total number of EMMs in a given rank. Once a given EMM has been selected and broadcast, it is moved to a second identical storage area in the RAM 3324, again using the hash function. Hence the first area diminishes in size as the EMMs are broadcast, to the extent that, once a complete rank has been used, this is deleted. Once the first storage area is completely empty, it is replaced by the second storage area before a new round of EMM broadcast, and vice versa.

In the above fashion, after two or three cycles of the EMMs, statistically the chances of any two EMMs destined for the same end user being transmitted back to back is negligible.

At regular intervals whilst the EMMs are being stored the computer 3050 computes the number of bytes in storage and from this computes the bitrate of emission given the global bitrate directive from the multiplexer and software multiplexer.

Reference was made above to the backup databases 3320 and 3322. These are in fact in the preferred embodiment sequential file stores, which hold a backup version of what is in the RAM 3324. In the event of failure of the Message Emitter and subsequent restart, or more generally when the ME is being restarted for whatever reason, a link is made between the RAM and the databases, over which the stored EMMs are uploaded to RAM. In this way, the risk of losing EMMs in the event of failure can be removed.

Similar storage of PPV EMMs occurs to that described above in relation to subscription EMMs, with the rank typically corresponding to a given operator and the number in the rank corresponding to the session number.

Smartcard

A daughter, or "subscriber", smartcard 3020 is schematically shown in Figure 18 and comprises an 8 bit microprocessor 110, such as a Motorola 6805 microprocessor, having an input/output bus coupled to a standard array of contacts 120 which in use are connected to a corresponding array of contacts in the card reader of the receiver/decoder 2020, the card reader being of conventional design. The microprocessor 110 is also provided with bus connections to preferably masked ROM 130, RAM 140 and EEPROM 150. The smartcard complies with the ISO 7816–1, 7816–2 and 7816–3 standard protocols which determine certain physical parameters of the smartcard, the positions of the contacts on the chip and certain communications between the external system (and particularly the receiver/decoder 2020) and the smartcard respectively and which will therefore not be further described here. One function of the microprocessor 110 is to manage the memory in the smartcard, as now described.

The EEPROM 150 contains certain dynamically–created operator zones 154, 155, 156 and dynamically–created data zones which will now be described with reference to Figure 19.

Referring to Figure 19, EEPROM 150 comprises a permanent "card ID" (or manufacturer) zone 151 of 8 bytes which contains a permanent subscriber smartcard identifier set by the manufacturer of the smartcard 3020.

When the smartcard is reset, the microprocessor 110 issues a signal to receiver/decoder 2020, the signal comprising an identifier of the conditional access system used by the smartcard and data generated from data stored in the smartcard, including the card ID. This signal is stored by the receiver/decoder 2020, which subsequently utilises the stored signal to check whether the smartcard is compatible with the conditional access system used by the receiver/decoder 2020.

The EEPROM 150 also contains a permanent "random number generator" zone 152 which contains a program for generating pseudo-random numbers. Such random numbers are used for diversifying transaction output signals generated by the smartcard 3020 and sent back to the broadcaster.

Below the random number generator zone 152 a permanent "management" zone 153 of 144 bytes is provided. The permanent management zone 153 is a specific operator zone utilised by a program in the ROM 130 in the dynamic creation (and removal) of zones 154, 155, 156... as described below. The permanent management zone 153 contains data relating to the rights of the smartcard to create or remove zones.

The program for dynamically creating and removing zones is responsive to specific zone creation (or removal) EMMs which are transmitted by the SAS 3002 and received by the receiver/decoder 2020 and passed to the subscriber smartcard 3020. In order to create the EMMs the operator requires specific keys dedicated to the management zone. This prevents one operator from deleting zones relating to another operator.

Below the management zone 153 is a series of "operator ID" zones 154, 155, 156 for operators 1, 2 ..... N respectively. Normally at least one operator ID zone will be preloaded into the EEPROM of the subscriber smartcard 3020 so that the end user can decrypt programmes broadcast by that operator. However further operator ID zones can subsequently be dynamically created using the management zone 153 in response to a transaction output signal generated via his smartcard 3020 by the end user (subscriber), as will subsequently be described.

Each operator zone 154, 155, 156 contains the identifier of the group to which the smartcard 3020 belongs, and the position of the smartcard within the group. This data enables the smartcard (along with the other smartcards in its group) to be responsive to a broadcast "group" subscription EMM having that group's address (but not the smartcard's position in the group) as well as to an "individual" (or commercial offers subscription) EMM addressed only to that smartcard within the group. There can be 256 member smartcards of each such group and this feature therefore reduces significantly the bandwidth required for broadcasting EMMs.

In order to reduce further the bandwidth required for broadcasting "group" subscription EMMs, the group data in each operator zone 154, 155, 156 and all similar zones in the EEPROM of smartcard 3020 and the other daughter smartcards is continually updated to enable a particular smartcard to change its position in each group to fill any holes created by e.g. deletion of a member of the group. The holes are filled by the SAS 3002 as in the STM server 3104 there is a list of such holes.

In this manner fragmentation is reduced and each group's membership is maintained at or near the maximum of 256 members.

Each operator zone 154, 155, 156 is associated with one or more "operator data objects" stored in the EEPROM 150. As shown in Figure 19, a series of dynamically created "operator data" objects 157–165 are located below the operator ID zones. Each of these objects is labelled with:

a) an "identifier" 1, 2, 3 .... N corresponding to its associated operator 1, 2, 3 ... N as shown in its left hand section in Figure 19;

b) an "ID" indicating the type of object; and

c) a "data" zone reserved for data, as shown in the right hand section of each relevant operator object in Figure 19. It should be understood that each operator is associated with a similar set of data objects so that the following description of the types of data in the data objects of operator 1 is also applicable to the data objects of all the other operators. Also it will be noted that the data objects are located in contiguous physical regions of the EEPROM and that their order is immaterial.

Deletion of a data object creates a "hole" 166 in the smartcard, that is, the number of bytes that the deleted objects had previously occupied are not immediately occupied. The thus "freed" number of bytes, or "hole" are labelled with:

a) an "identifier" 0; and

b) an "ID" indicating that the bytes are free to receive an object.

The next data object created fills the hole, as identified by the identifier 0. In this manner the limited memory capacity (4 kilobytes) of the EEPROM 150 is efficiently utilised.

Turning now to the set of data objects associated with each operator, examples of the data objects are now described.

Data object 157 contains an EMM key used for decrypting encrypted EMM's received by the receiver/decoder 2020. This EMM key is permanently stored in the data object 157. This data object 157 may be created prior to distribution of the smartcard 3020, and/or may be created dynamically when creating a new operator zone (as described above).

Data object 159 contains ECM keys which are sent by the associated operator (in this case operator 1) to enable the end user to decrypt the particular "bouquet" of programs to which he has subscribed. New ECM keys are sent typically every month, along with a group subscription (renewal) EMM which renews the end user's overall right to view the broadcast from (in this case) operator 1. The use of separate EMM and ECM keys enables viewing rights to be purchased in different ways (in this embodiment by subscription and individually (Pay Per View)) and also increases security. The Pay Per View (PPV) mode will be described subsequently.

Since new ECM keys are sent periodically, it is essential to prevent a user from usin old ECM keys, for example by switching off the receiver/decoder or re-setting a clock to prevent expiry of an old ECM key so that a timer in the receiver/decoder 2020 could be overridden. Accordingly operatot zone 154 comprises an area (typically having a size of 2 bytes) containing an obsolescence date of the ECM keys. The smartcard 3020 is arranged to compare this date with the current date which is contained in received ECMs and to prevent decryption if the current date is later than the obsolescence date. The obsolescence date is transmitted via EMMs, as described above.

Data object 161 contains a 64 bit subscription bitmap which is an exact representation of the broadcast operator's programs to which the subscriber has subscribed. Every bit represents a program and is set to "1" if it is subscribed to and "0" if it is not.

Data object 163 contains a quantity of tokens which can be used by the consumer in PPV mode to buy viewing rights to an imminent broadcast e.g. in response to a free preview or other advertisement. Data object 163 also contains a limit value, which may be set to e.g. a negative value to allow credit to the consumer. Tokens can be purchased e.g. by credit and via the modemmed back channel 4002, or by using a voice server in combination with a credit card, for example. A particular event can be charged as one token or a number of tokens.

Data object 165 contains a description of a PPV event, as shown with reference to table 167 of Figure 20.

The PPV event description 167 contains a "session ID" 168 identifying the viewing session (corresponding to the program and the time and date of broadcasting) a "session mode" 169 indicating how the viewing right is being purchased (e.g. in pre-book mode), a "session index" 170 and a "session view" 171.

In respect of receiving a programme in PPV mode, the receiver decoder 2020 determines whether the programme is one sold in PPV mode. If so, the decoder 2020 checks, using the items stored in the PPV event description 167 whether the session ID for the programme is stored therein. If the session ID is stored therein, the control word is extracted from the ECM.

If the session ID is not stored therein, by means of a specific application the receiver/decoder 2020 displays a message to the end user indicating that he has the right to view the session at a cost of, say, 25 tokens, as read from the ECM or to connect to the communications servers 3022 to purchase the event. Using the tokens, if the end user answers "yes" (by means of remote controller 2026 (see Figure 2)) the decoder 2020 sends the ECM to the smartcard, the smartcard decreases the wallet of the smartcard 3020 by 25 tokens, writes the session ID 168, the session mode 169, the session index 170 and the session view 171 in the PPV event description 167 and extracts and deciphers the control word from the ECM.

In the "pre-book" mode, an EMM will be passed to the smartcard 3020 so that the smartcard will write the session ID 168, the session mode 169, the session index 170 and the session view 171 in the PPV event description 167 using the EMM.

The session index 170 can be set to differentiate one broadcast from the other. This feature permits authorization to be given for a subset of broadcasts, for example, 3 times out of 5 broadcasts. As soon as an ECM with a session index different from the current session index 170 stored in the PPV event description 167 is passed to the smartcard, the number of the session view 171 is decreased by one. When the session view reaches zero, the smartcard will refuse to decipher an ECM with a different session index to the current session index.

The initial value of the session view depends only on the way in which the broadcast supplier wishes to define the event to which it relates; the session view for a respective event may take any value.

The microprocessor 110 in the smartcard implements a counting and a comparison program to detect when the limit to the number of viewings of a particular program has been reached.

All of the session ID 168, the session mode 169, the session index 170 and the session view 171 in the PPV event description 167 may be extracted from the smartcard using the "call-back" procedure as described previously.

Each receiver/decoder 2020 contains an identifier which may either identify uniquely that receiver/decoder or identify its manufacturer or may classify it in some other way in order to enable it to work only with a particular individual smartcard, a particular class of smartcards made by the same or a corresponding manufacturer or any other class of smartcards which are intended for use with that class of receiver/decoders exclusively.

In this manner the receiver/decoders 2020 which have been supplied by one broadcast supplier to the consumer are protected against the use of non-authorised daughter smartcards 3020.

Additionally or alternatively to this first "handshake" between the smartcard and the receiver, the EEPROM of the smartcard 3020 could contain a field or bitmap describing the categories of receiver/decoders 2020 with which it can function. These could be specified either during the manufacture of the smartcard 3020 or by a specific EMM.

The bitmap stored in the smartcard 3020 typically comprises a list of up to 80 receiver/decoders, each identified with a corresponding receiver/decoder ID with which the smartcard may be used. Associated with each receiver/decoder is a level "1" or "0" indicating whether the smartcard may be used with the receiver/decoder or not, respectively. A program in the memory 2024 of the receiver/decoder searches for the identifier of the receiver/decoder in the bitmap stored in the smartcard. If the identifier is found, and the value associated with the identifier is "1", then the smartcard is "enabled"; if not, then the smartcard will not function with that receiver/decoder.

In addition, if, typically because of an agreement between operators, it is desired to authorize the use of other smartcards in a particular receiver/decoder, specific EMMs will be sent to those smartcards to change their bitmap via the transponder 2014.

Each broadcast supplier may differentiate his subscribers according to certain predetermined criteria. For example, a number of subscribers may be classed as "VIPs". Accordingly, each broadcast supplier may divide his subscribers into a plurality of subsets, each subset comprising any number of subscribers.

The subset to which a particular subscriber belongs is set in the SMS 3004. In turn, the SAS 3002 transmits an EMM to the subscriber which writes information (typically of length 1 byte) concerning the subset to which the subscriber belongs into the relevant operator data zone, say 154, of the EEPROM of the smartcard. In turn, as events are broadcast by the broadcast supplier, an ECM, typically of 256 bits, is transmitted with the event and indicating which of the subsets of subscribers may view the event. If, according to the information stored in the operator zone, the subscriber does not have the right to view the event, as determined by the ECM, programme viewing is denied.

This facility may be used, for example, to switch off all of a given operator's smartcards in a particular geographical region during the transmission of a particular program, in particular a program relating to a sports fixture taking place in that geographical region. In this manner football clubs and other sport bodies can sell broadcasting rights outside their locality whilst preventing local supporters from viewing the fixture on television. In this manner the local supporters are encouraged to buy tickets and attend the fixture.

Each of the features associated with zones 151 to 172 is considered to be a separate invention independent of the dynamic creation of zones.

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.

Each feature disclosed in the description, and (where appropriate) the claims and drawings may be provided independently of in any appropriate combination.

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.

Cross reference is made to our co-pending applications, all bearing the same filing date, and entitled Signal Generation and Broadcasting (Attorney Reference no. PC/ASB/19707), Smartcard for use with a Receiver of Encrypted Broadcast Signals, and Receiver (Attorney Reference No. PC/ASB/19708), Broadcast and Reception System and Conditional Access System therefor (Attorney Reference No. PC/ASB/19710), Downloading a Computer File from a Transmitter via a Receiver/Decoder to a Computer (Attorney Reference No. PC/ASB/19711), Transmission and Reception of Television Programmes and Other Data (Attorney Reference No. PC/ASB/19712), Downloading Data (Attorney Reference No. PC/ASB/19713), Computer Memory Organisation (Attorney Reference No. PC/ASB/19714), Television or Radio Control System Development (Attorney Reference No. PC/ASB/19715), Extracting Data Sections from a Transmitted Data Stream (Attorney Reference No. PC/ASB/19716), Access Control System (Attorney Reference No. PC/ASB/19717), Data Processing System (Attorney Reference No. PC/ASB/19718), and Broadcast and Reception System, and Receiver/Decoder and Remote Controller therefor (Attorney Reference No. PC/ASB/19720). The disclosures of these documents are incorporated herein by reference. The list of applications includes the present application.

CLAIMS

1. A conditional access system comprising:

means for generating a plurality of messages; and

means for receiving the messages, said receiving means being adapted to communicate with said generating means via a communications server connected directly to said generating means.

2. A conditional access system according to Claim 1, wherein said message is an entitlement message for transmission to the receiving means, said generating means being adapted to generate entitlement messages in response to data received from said receiving means.

3. A conditional access system according to Claim 1 or 2, wherein said generating means is arranged to transmit a message as a packet of digital data to said receiving means either via said communications server or via a satellite transponder.

4. A conditional access system according to any preceding claim, wherein said receiving means is connectable to said communications server via a modem and telephone link.

5. A conditional access system for affording conditional access to subscribers, comprising:

a subscriber management system;

a subscriber authorization system coupled to the subscriber management system; and

a communications server; said server being connected directly to the subscriber authorization system.

6. A conditional access system according to Claim 5, further comprising a receiver/decoder for the subscriber, the receiver/decoder being connectable to said communications server, and hence to said subscriber authorization system, via a

modem and telephone link.

7. A broadcast and reception system including a conditional access system according to any preceding claim.

8. A broadcast and reception system comprising:

means for generating a plurality of entitlement messages relating to broadcast programs;

means for receiving said messages from said generating means; and

means for connecting the receiving means to the generating means to receive said messages, said connecting means being capable of effecting a dedicated connection between the receiving means and the generating means.

9. A broadcast and reception system comprising:

means for generating a plurality of entitlement messages relating to broadcast programs;

means for receiving said messages from said generating means via a modem; and

means for connecting said modem to said generating means and said receiving means.

10. A broadcast and reception system according to Claim 9, wherein said generating means is connected to said modem via a communications server.

11. A broadcast and reception system according to Claim 9 or 10, wherein said receiving means is adapted to communicate with said generating means via said modem and connecting means.

12. A broadcast and reception system according to Claim 11, wherein said receiving means comprises means for reading a smartcard insertable thereinto by an end user, the smartcard having stored therein data to initiate automatically the transmission of a message from said receiving means to said generating means upon

insertion of the smartcard by the end user.

13. A broadcast and reception system according to Claim 11or 12, further comprising a voice link to enable the end user of the broadcast and reception system to communicate with the generating means.

14. A broadcast and reception system according to any of Claims 8 to 13, wherein said receiving means comprises a receiver/decoder comprising means for receiving a compressed MPEG–type signal, means for decoding the received signal to provide a television signal and means for supplying the television signal to a television.

15. A broadcast and reception system, comprising, at the broadcast end:

a broadcast system including means for broadcasting a callback request;

and at the reception end:

a receiver including means for calling back the broadcast system in response to the callback request.

16. A system according to Claim 15, wherein the means for calling back the broadcast system includes a modem connectable to a telephone system.

17. A system according to Claim 15 or 16, wherein the means for calling back the broadcast system is arranged to transfer to the broadcast system information concerning the receiver.

18. A system according to Claim 17, wherein the broadcast system includes means for storing the information.

19. A system according to any of Claims 15 to 18, wherein the broadcast means is arranged to broadcast a callback request which includes a command that the callback be made at a given time, and the means for calling back the broadcast system is arranged to respond to said command.

20. A system according to any of Claims 15 to 19, wherein the broadcasting means is arranged to broadcast as the callback request one or more entitlement messages for broadcast.

21. A system according to any of Claims 15 to 20, wherein the broadcast system includes means for generating a check message and passing this to the receiver, the receiver includes means for encrypting the check message and passing this to the broadcast system, and the broadcast system further includes means for decrypting the check message received from the receiver and comparing this with the original check message.

22. A conditional access system or a broadcast and reception system substantially as herein described with reference to and as illustrated in the accompanying drawings, and especially Figures 12, 13 or 14 thereof.

Fig.1.

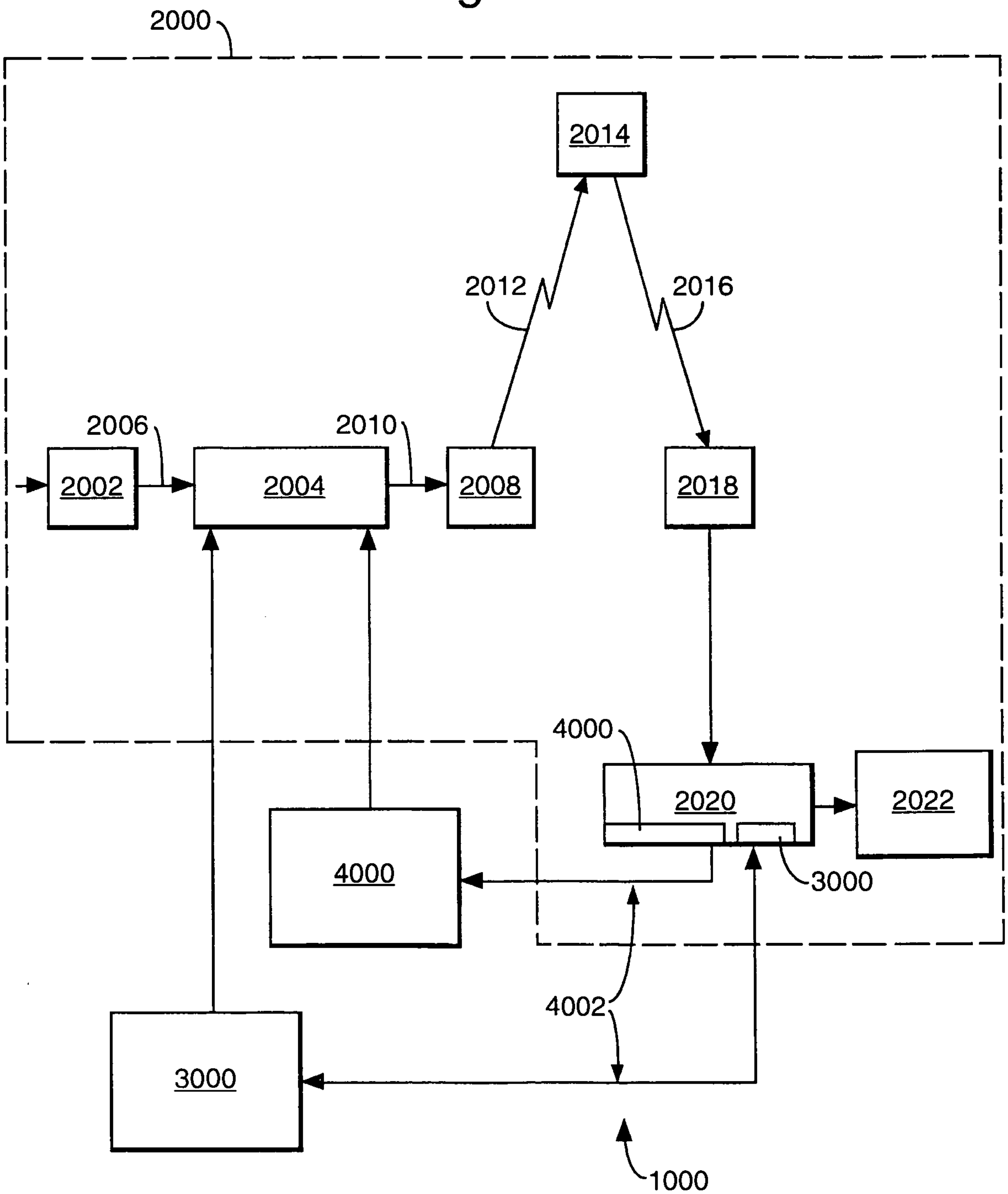

2000
2014
2012
2016
2006
2010
2002
2004
2008
2018
4000
2020
2022
4000
3000
4002
3000
1000

Fig.2.

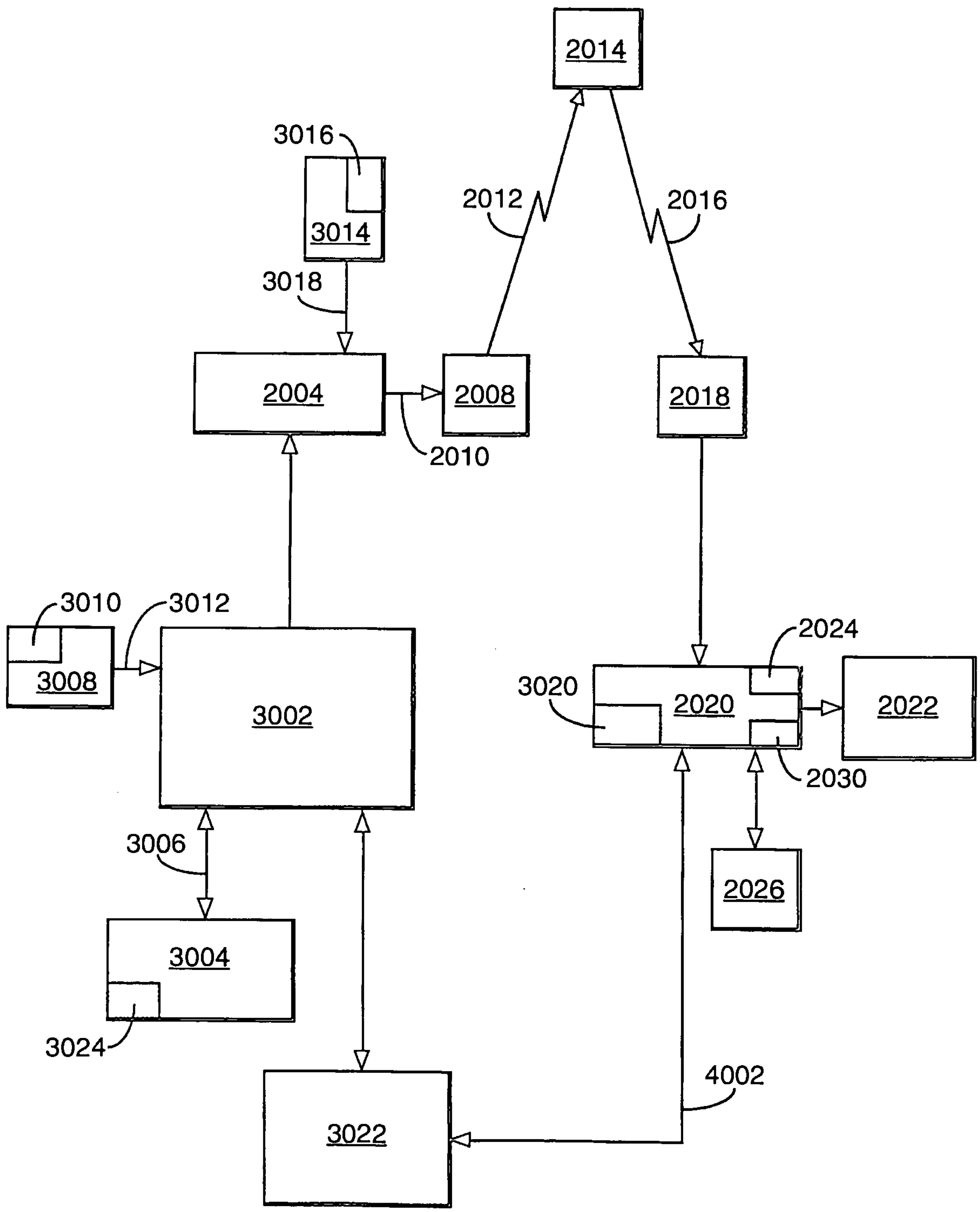
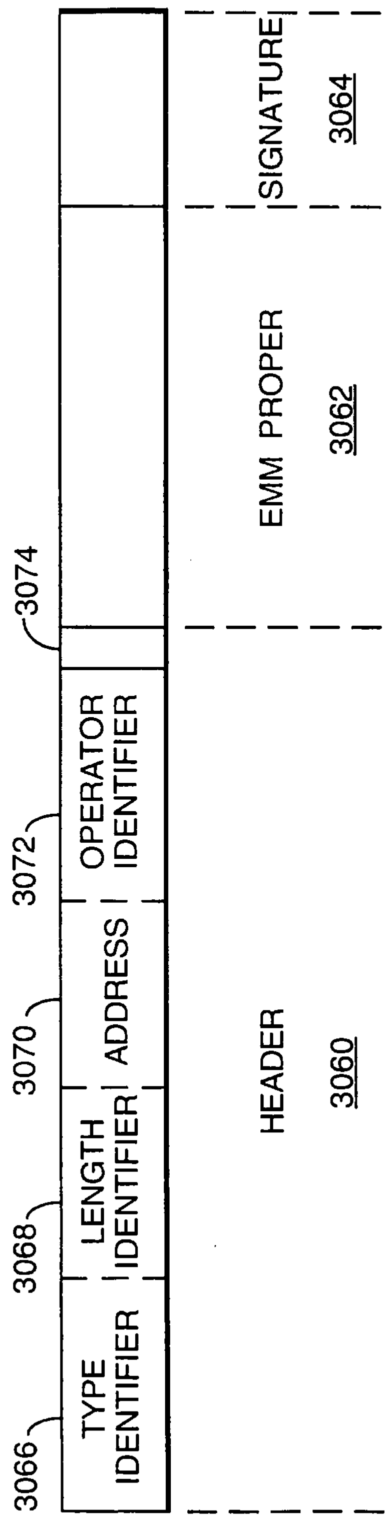

2014
3016
2012
2016
3014
3018
2004
2008
2018
2010
3010
3012
2024
3008
3002
3020
2020
2022
2030
3006
2026
3004
3024
4002
3022

Fig.3.


TYPE IDENTIFIER
LENGTH IDENTIFIER
ADDRESS
OPERATOR IDENTIFIER
3066
3068
3070
3072
3074
HEADER
3060
EMM PROPER
3062
SIGNATURE
3064

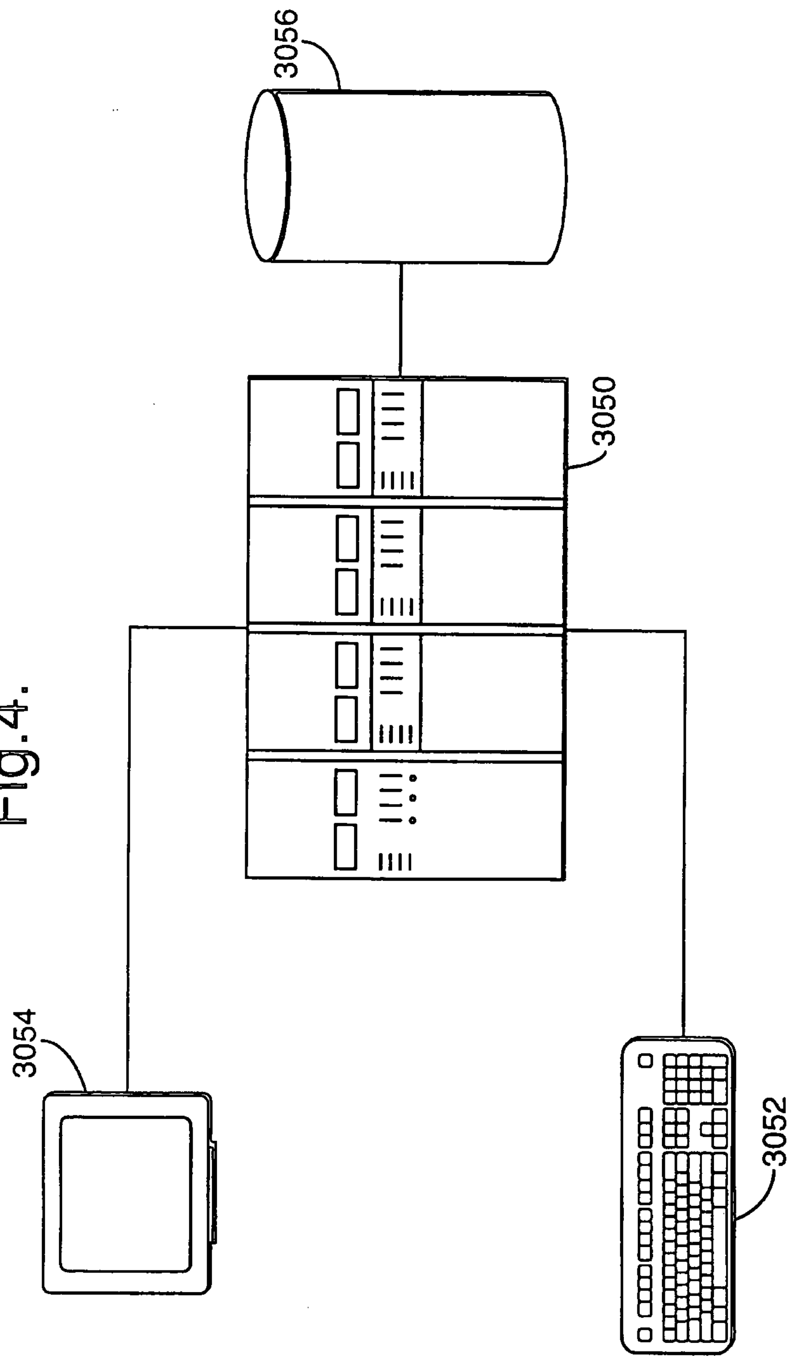
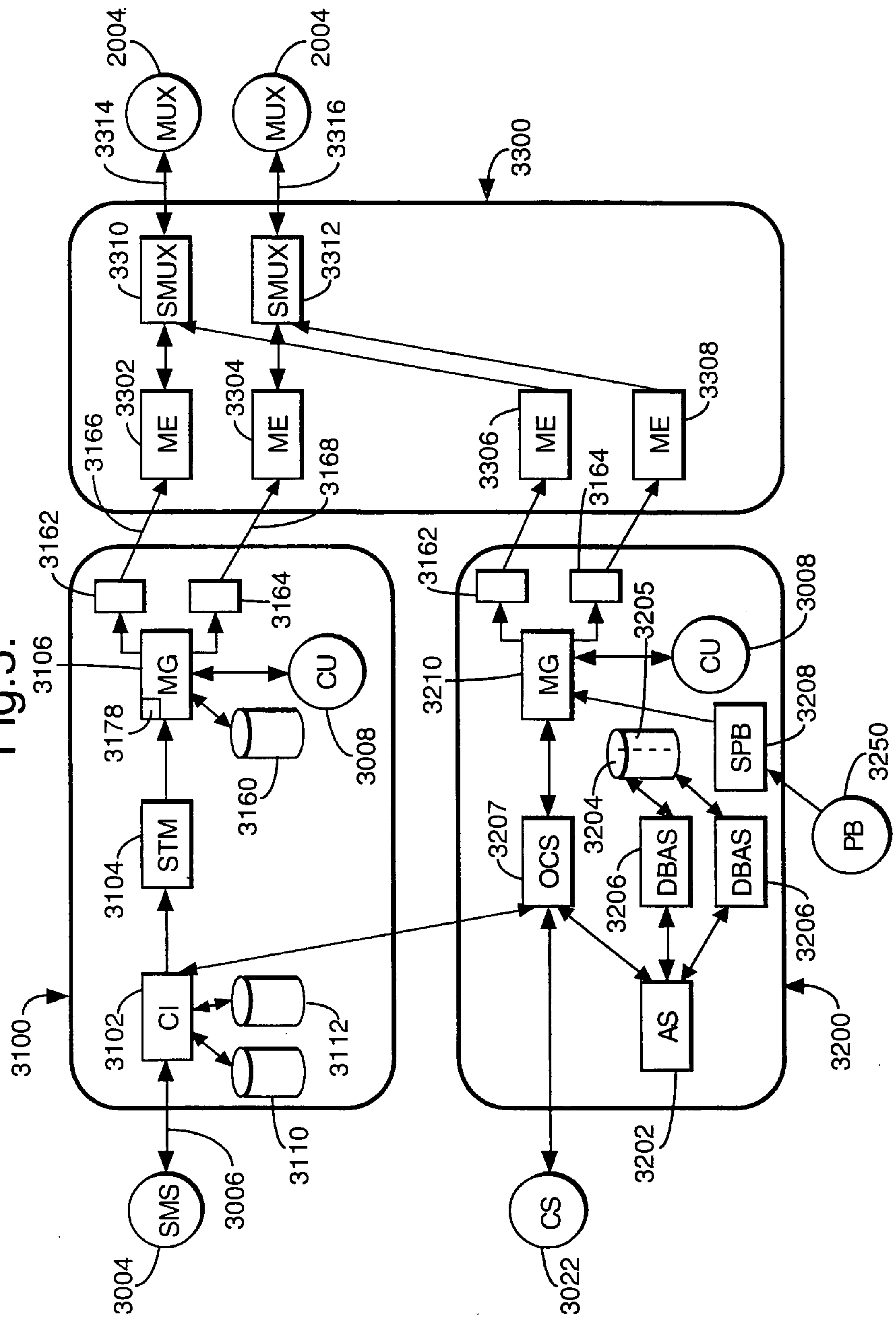


Fig.4.

Fig.5.



SMS
3004
3006
CI
3102
3110
3112
STM
3104
MG
3106
3178
3160
CU
3008
3162
3164
3100
CS
3022
AS
3202
OCS
3207
DBAS
3206
3204
3205
SPB
3208
PB
3250
3210
3200
ME
3302
3304
3306
3308
3166
3168
SMUX
3310
3312
MUX
2004
3314
3316
3300

Fig.6.

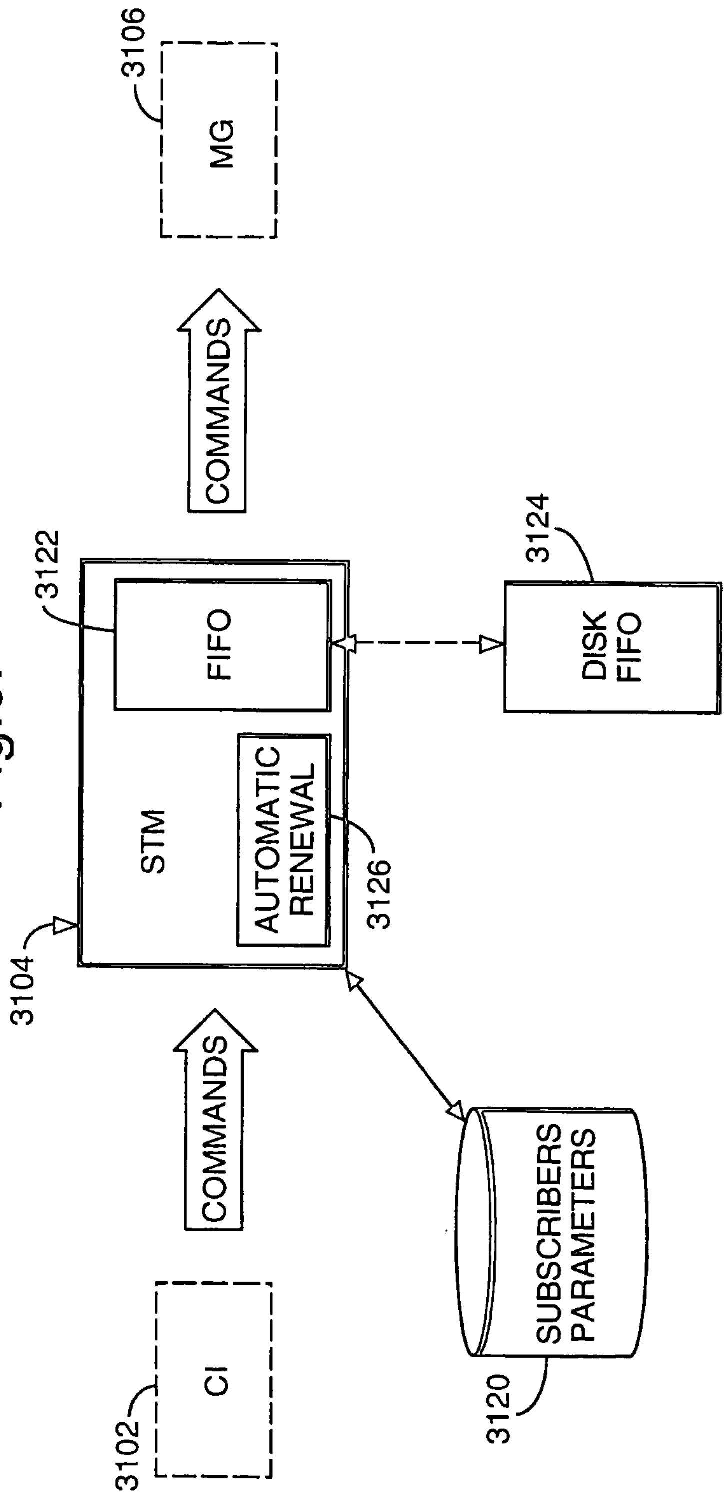

3106
MG
COMMANDS
3122
FIFO
3124
DISK
FIFO
STM
AUTOMATIC
RENEWAL
3126
3104
COMMANDS
SUBSCRIBERS
PARAMETERS
3120
CI
3102

Fig.7.

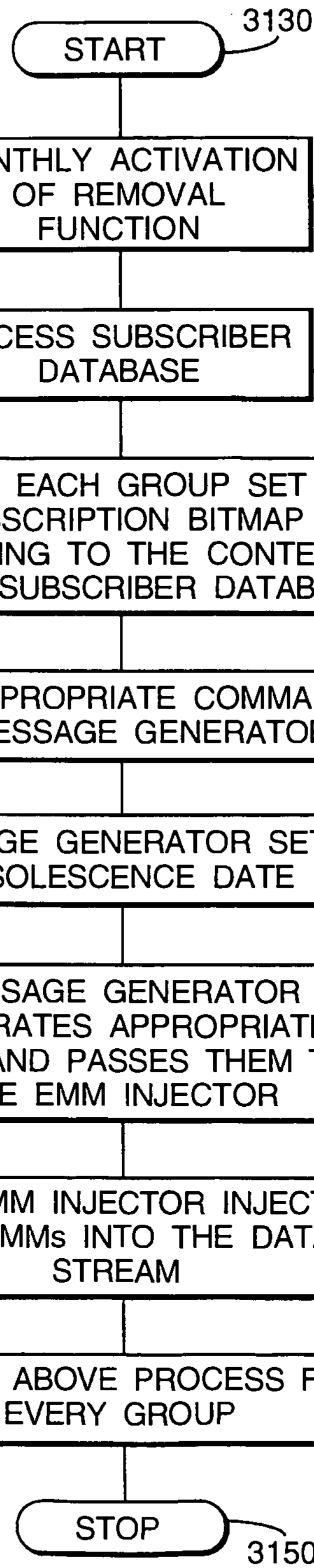

START
3130
MONTHLY ACTIVATION OF REMOVAL FUNCTION
3132
ACCESS SUBSCRIBER DATABASE
3134
FOR EACH GROUP SET SUBSCRIPTION BITMAP ACCORDING TO THE CONTENTS OF THE SUBSCRIBER DATABASE
3136
PASS APPROPRIATE COMMANDS TO MESSAGE GENERATOR
3142
MESSAGE GENERATOR SETS OBSOLESCENCE DATE
3143
MESSAGE GENERATOR GENERATES APPROPRIATE EMMs AND PASSES THEM TO THE EMM INJECTOR
3144
THE EMM INJECTOR INJECTS THE EMMs INTO THE DATA STREAM
3146
REPEAT ABOVE PROCESS FOR EVERY GROUP
3148
STOP
3150

Fig.8.

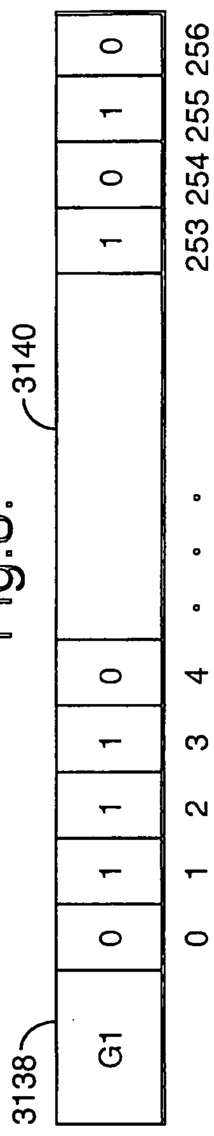


Fig.9.

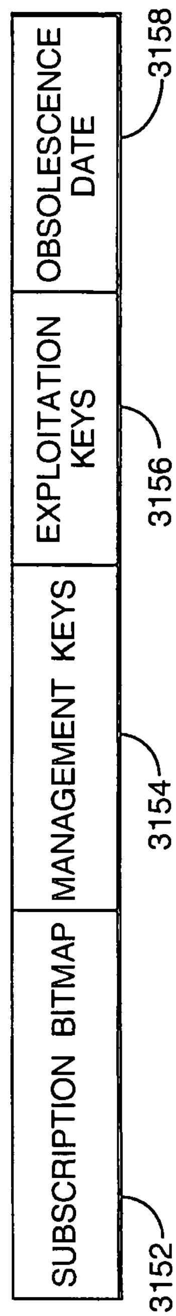


Fig.10.

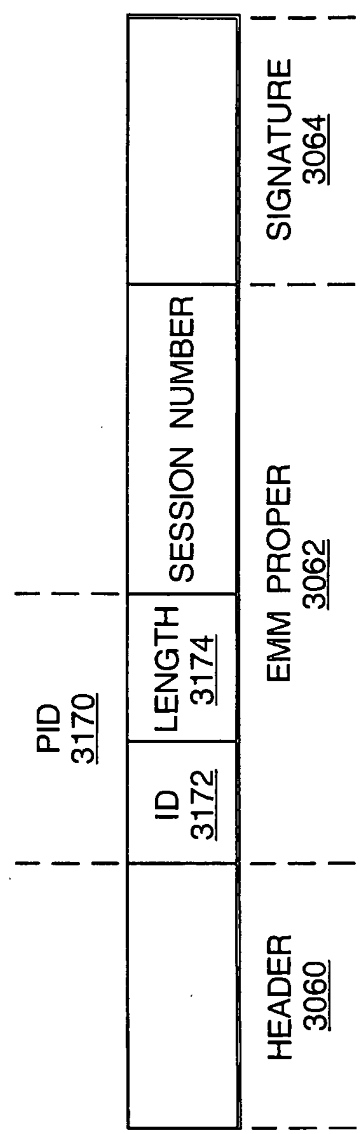


Fig.11.

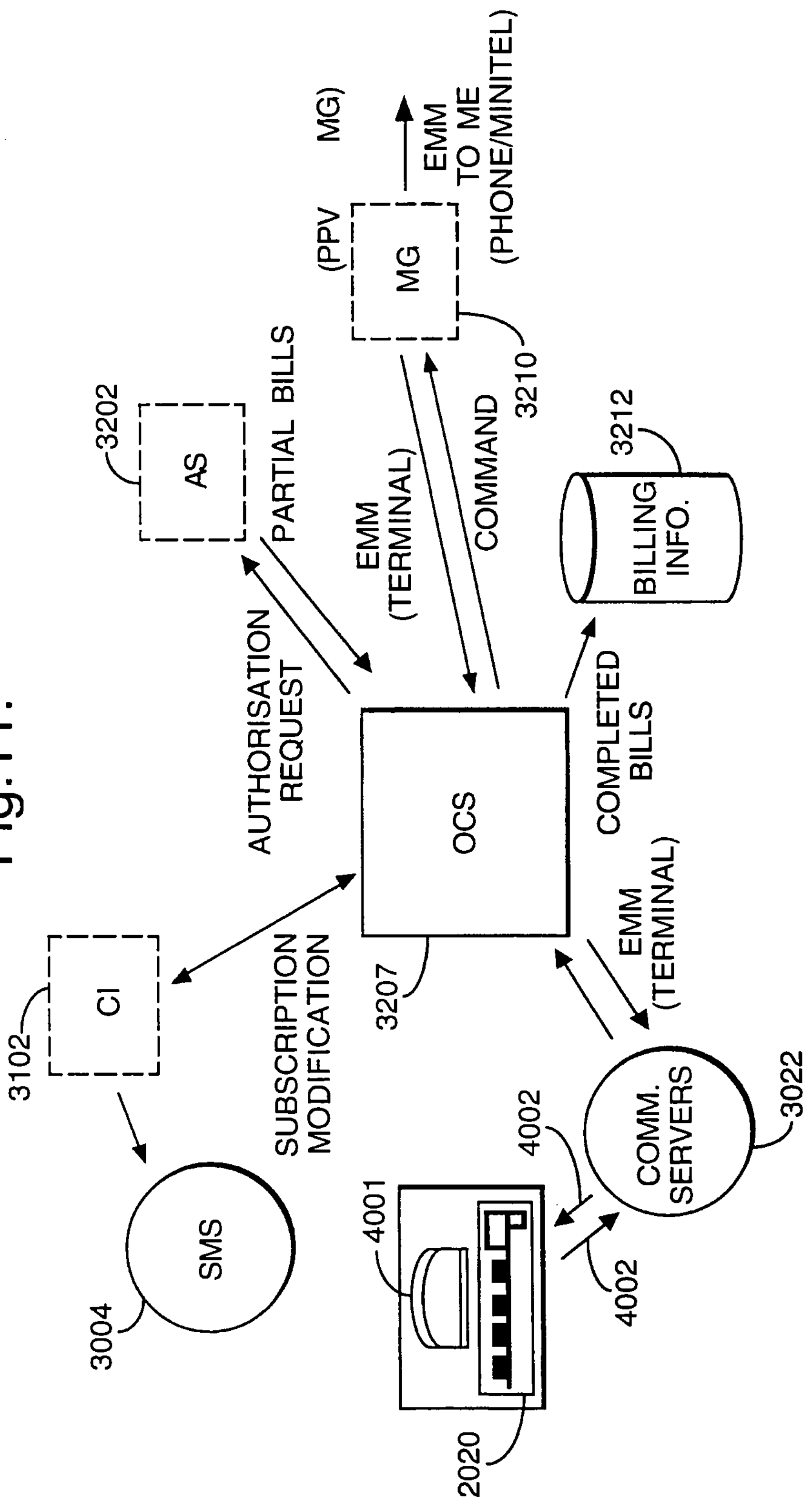
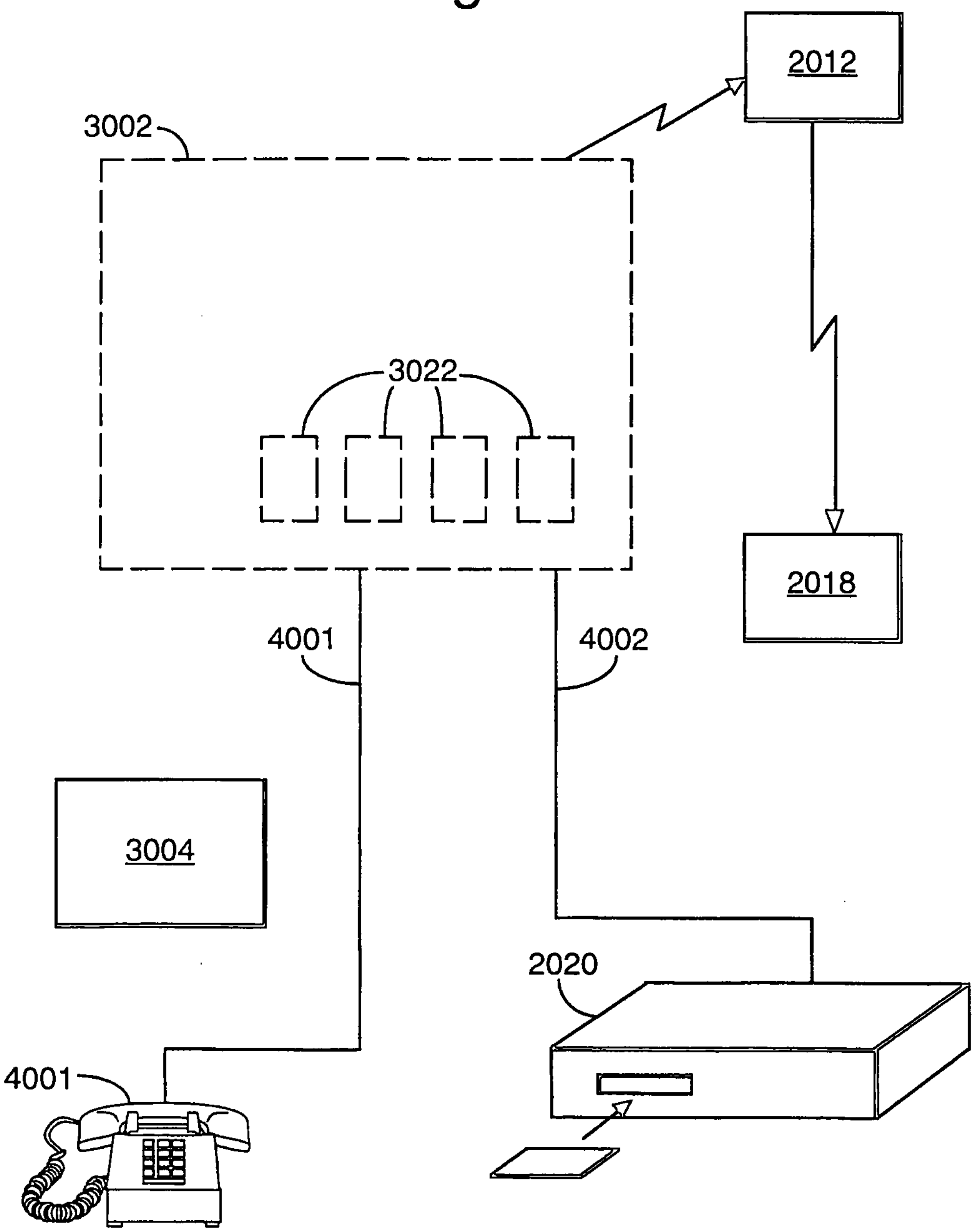

SMS
3004
CI
3102
SUBSCRIPTION
MODIFICATION
AS
3202
AUTHORISATION
REQUEST
PARTIAL BILLS
OCS
3207
(PPV MG)
MG
3210
EMM
TO ME
(PHONE/MINITEL)
EMM
(TERMINAL)
COMMAND
BILLING
INFO.
3212
COMPLETED
BILLS
EMM
(TERMINAL)
COMM.
SERVERS
3022
4002
4002
4001
2020

Fig.12.



SUBSTITUTE SHEET (RULE 26)

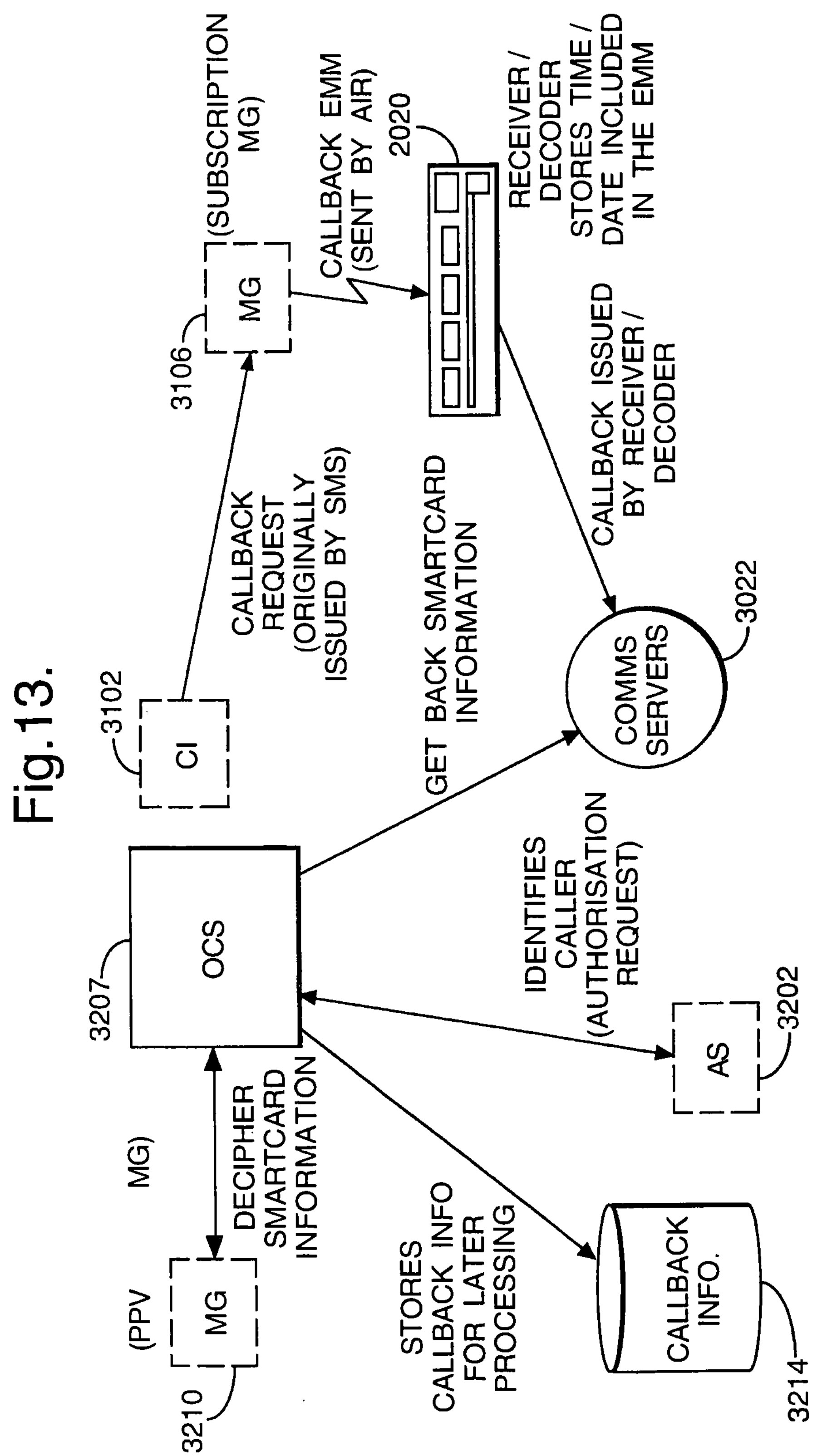
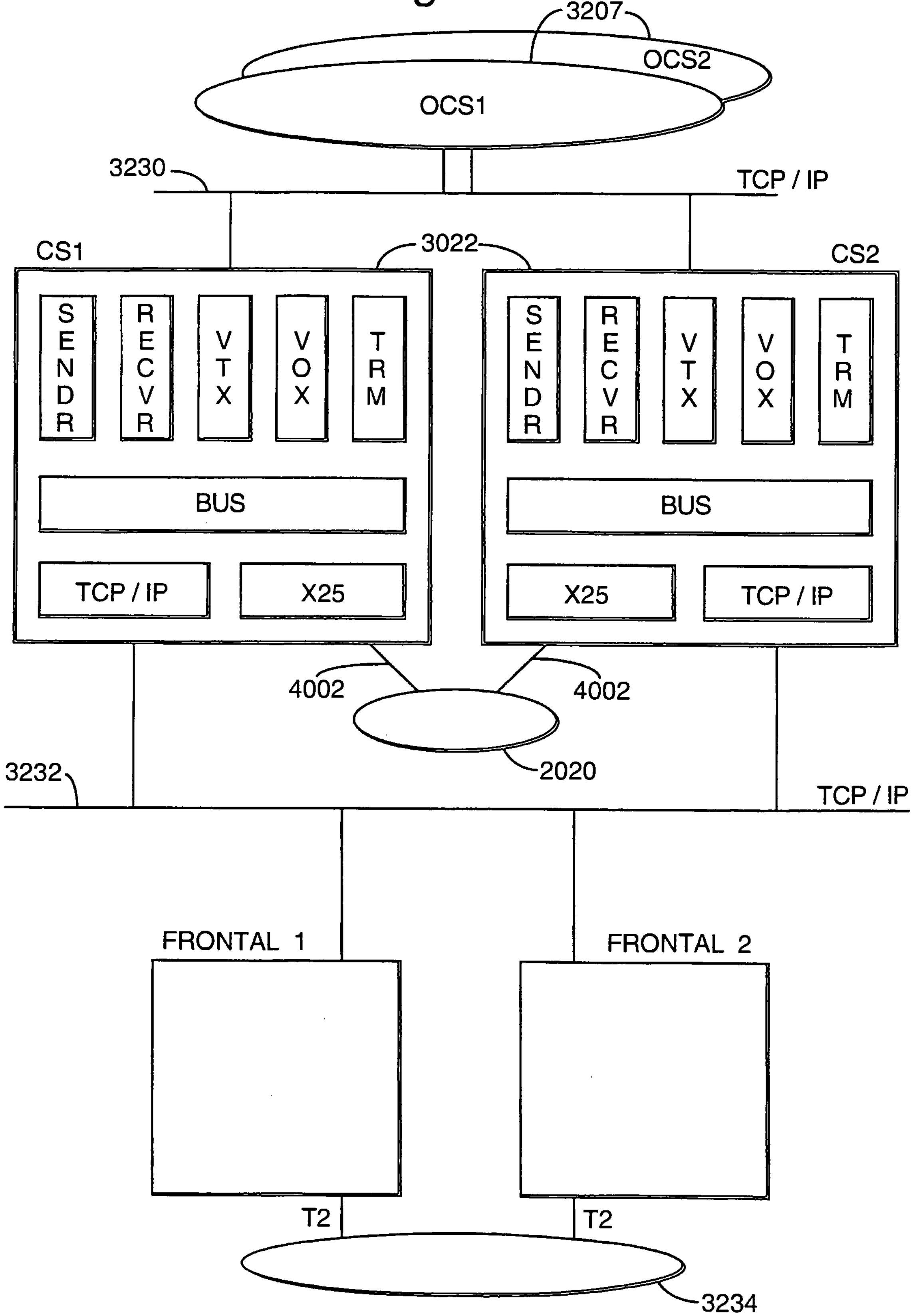

OCS
3207
(PPV MG)
MG
3210
DECIPHER SMARTCARD INFORMATION
STORES CALLBACK INFO FOR LATER PROCESSING
CALLBACK INFO.
3214
IDENTIFIES CALLER (AUTHORISATION REQUEST)
AS
3202
GET BACK SMARTCARD INFORMATION
COMMS SERVERS
3022
CI
3102
CALLBACK REQUEST (ORIGINALLY ISSUED BY SMS)
MG
3106
(SUBSCRIPTION MG)
CALLBACK EMM (SENT BY AIR)
2020
RECEIVER / DECODER STORES TIME / DATE INCLUDED IN THE EMM
CALLBACK ISSUED BY RECEIVER / DECODER

Fig.13.

Fig.14.


3207
OCS2
OCS1
3230
TCP / IP
CS1
3022
CS2
SENDR
RECVR
VTX
VOX
TRM
BUS
TCP / IP
X25
SENDR
RECVR
VTX
VOX
TRM
BUS
X25
TCP / IP
4002
4002
2020
3232
TCP / IP
FRONTAL 1
FRONTAL 2
T2
T2
3234

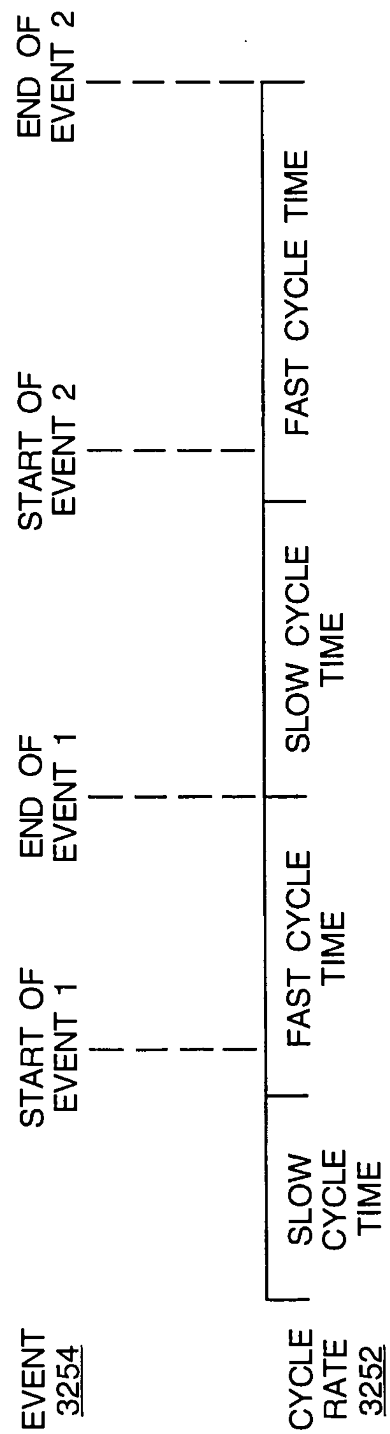

EVENT 3254
CYCLE RATE 3252
START OF EVENT 1
END OF EVENT 1
START OF EVENT 2
END OF EVENT 2
SLOW CYCLE TIME
FAST CYCLE TIME
SLOW CYCLE TIME
FAST CYCLE TIME

Fig.15.

Fig.16.

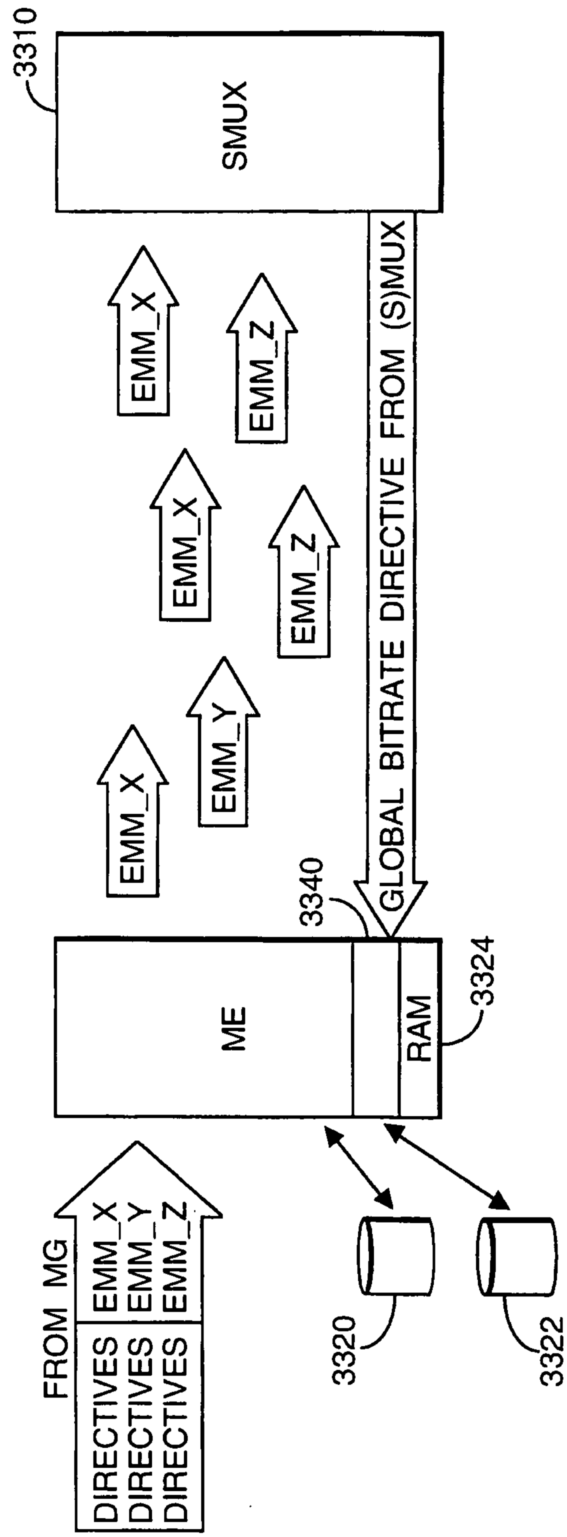

3310
SMUX
EMM_X
EMM_Z
EMM_X
EMM_Z
EMM_Y
EMM_X
GLOBAL BITRATE DIRECTIVE FROM (S)MUX
3340
ME
RAM
3324
FROM MG
DIRECTIVES EMM_X
DIRECTIVES EMM_Y
DIRECTIVES EMM_Z
3320
3322

Fig.17.

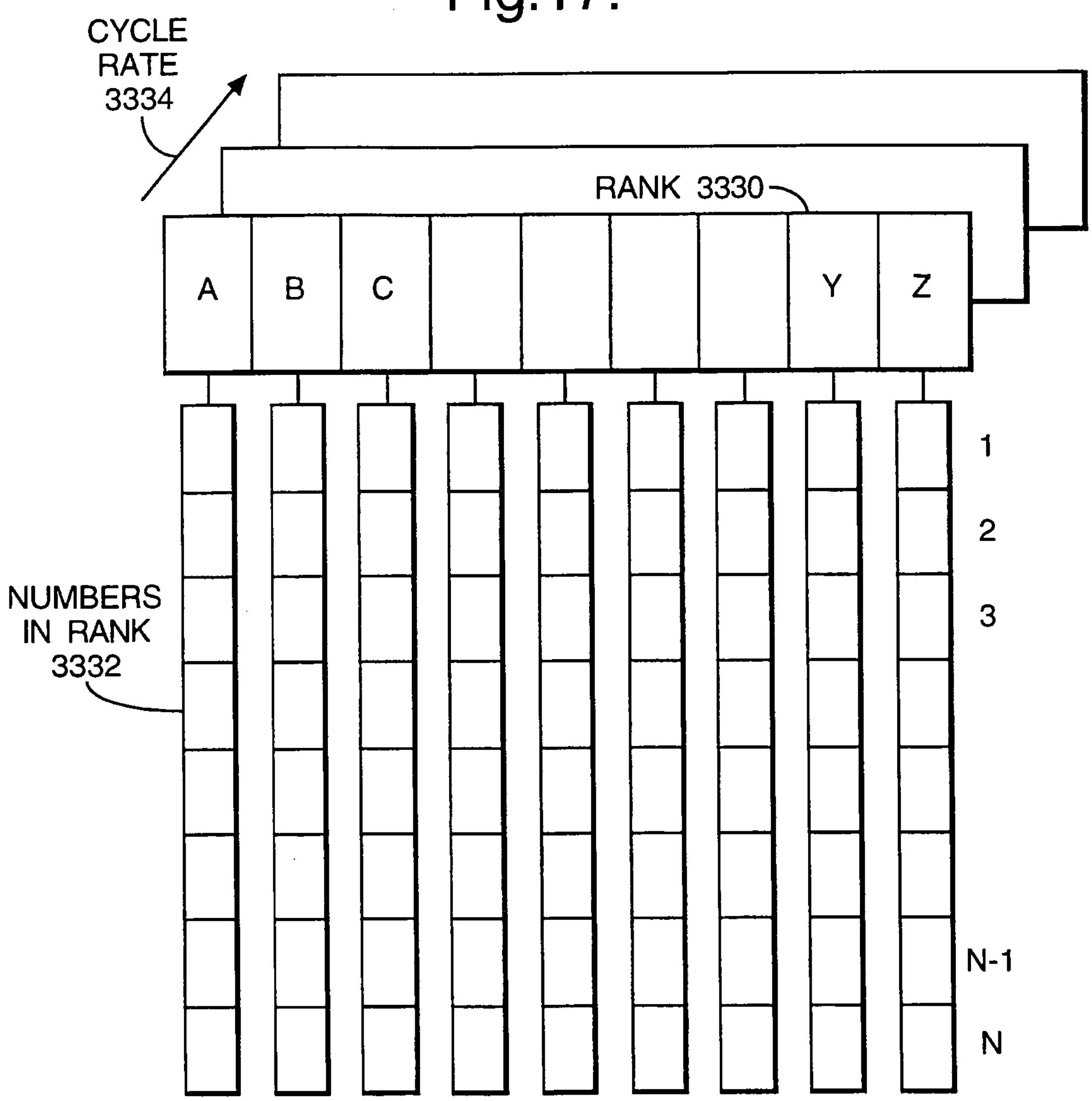


Fig.18.

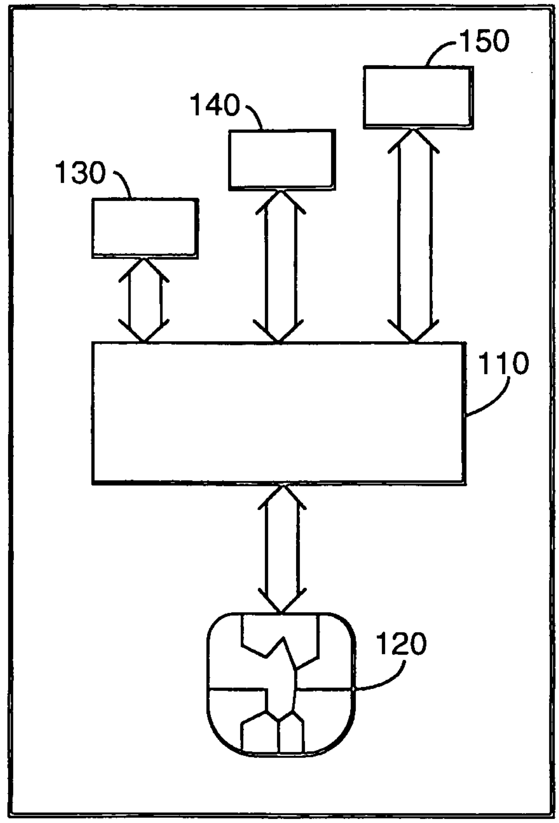

150
140
130
110
120

Fig.20.

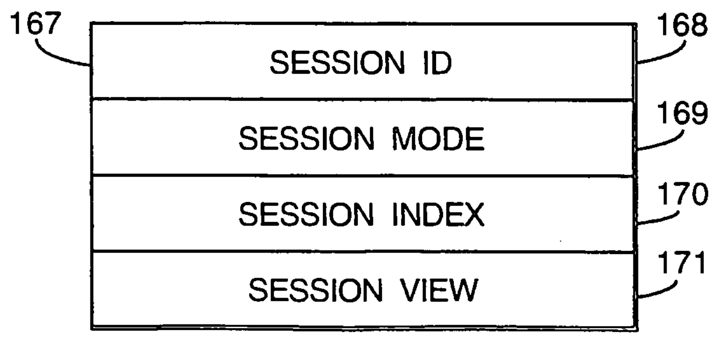
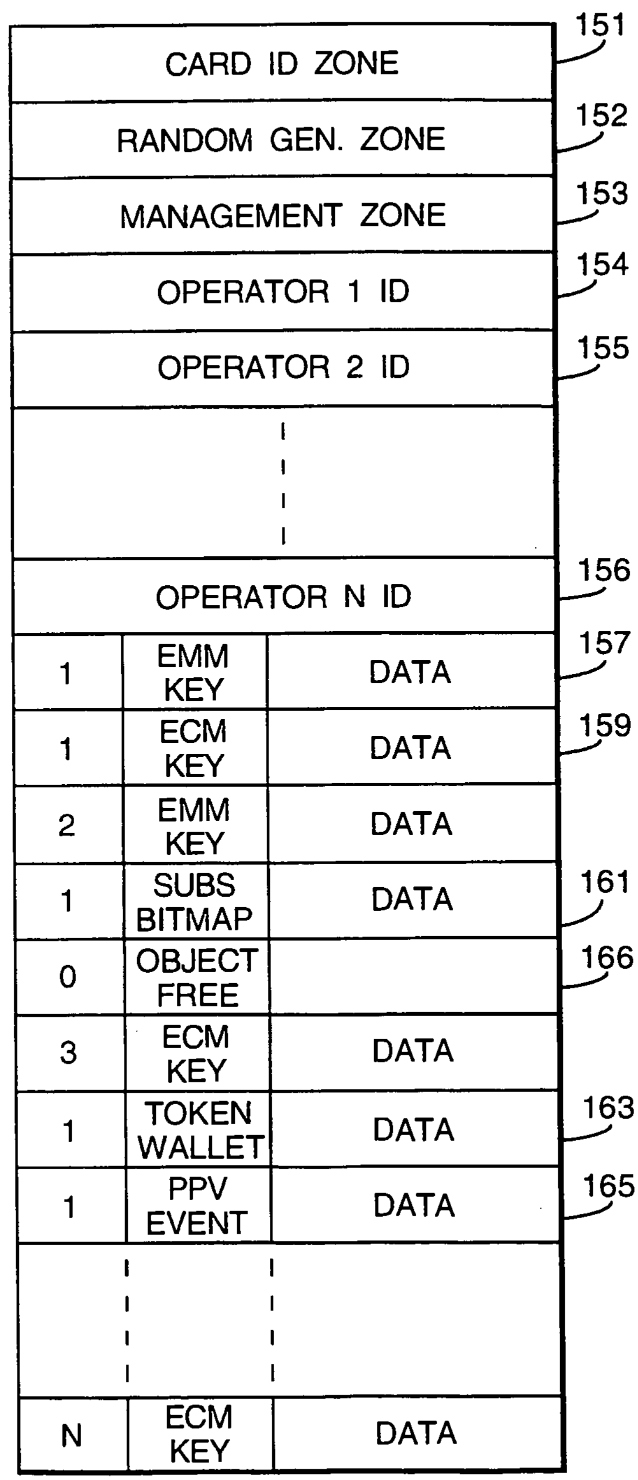

167
168
SESSION ID
169
SESSION MODE
170
SESSION INDEX
171
SESSION VIEW

Fig.19.


CARD ID ZONE
151
RANDOM GEN. ZONE
152
MANAGEMENT ZONE
153
OPERATOR 1 ID
154
OPERATOR 2 ID
155
OPERATOR N ID
156
1
EMM KEY
DATA
157
1
ECM KEY
DATA
159
2
EMM KEY
DATA
1
SUBS BITMAP
DATA
161
0
OBJECT FREE
166
3
ECM KEY
DATA
1
TOKEN WALLET
DATA
163
1
PPV EVENT
DATA
165
N
ECM KEY
DATA

INTERNATIONAL SEARCH REPORT

Intern al Application No
PCT/EP 97/02108

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 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 6 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	"FUNCTIONAL MODEL OF A CONDITIONAL ACCESS SYSTEM" EBU REVIEW- TECHNICAL, no. 266, 21 December 1995, pages 64-77, XP000559450 see the whole document ---	1-12, 15-19, 21,22
X	WO 94 14284 A (DISCOVERY COMMUNICAT INC) 23 June 1994 see page 8, line 8 - page 14, line 23 see page 18, line 28 - page 21, line 19 see page 24, line 25 - page 29, line 31 see page 33, line 8 - line 17 see figures 1-11 --- -/--	1-12, 14-17

[X] Further documents are listed in the continuation of box C.

[X] 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

1

Date of the actual completion of the international search	Date of mailing of the international search report
11 November 1997	18. 11. 97
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

Interna ıl Application No
PCT/EP 97/02108

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 144 663 A (KUDELSKI ANDRE ET AL) 1 September 1992 see column 2, line 5 - line 23 see column 3, line 6 - column 4, line 65 see column 5, line 62 - column 8, line 58 see figures 1-11 -----	1-12

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/EP 97/02108

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9414284 A	23-06-94	AU 5732994 A	04-07-94
		AU 5733094 A	04-07-94
		AU 5733194 A	04-07-94
		AU 5733294 A	04-07-94
		AU 5736394 A	04-07-94
		AU 5845894 A	22-06-94
		AU 5869894 A	04-07-94
		CA 2151458 A	23-06-94
		CN 1093211 A	05-10-94
		CN 1090451 A	03-08-94
		CN 1090452 A	03-08-94
		CN 1096151 A	07-12-94
		CN 1090453 A	03-08-94
		CN 1090454 A	03-08-94
		EP 0673578 A	27-09-95
		EP 0673579 A	27-09-95
		EP 0673580 A	27-09-95
		EP 0673581 A	27-09-95
		EP 0673582 A	27-09-95
		EP 0673583 A	27-09-95
		EP 0674824 A	04-10-95
		IL 107908 A	10-01-97
		IL 107909 A	15-04-97
		IL 107910 A	10-06-97
		IL 107912 A	18-02-97
		IL 107913 A	15-04-97
		JP 8510869 T	12-11-96
		JP 8506938 T	23-07-96
		JP 8506939 T	23-07-96
		JP 8506940 T	23-07-96
		JP 8506941 T	23-07-96
		JP 8506942 T	23-07-96
		NZ 259146 A	26-05-97
		NZ 259147 A	26-05-97
		NZ 259148 A	26-11-96
		WO 9413107 A	09-06-94
		WO 9414279 A	23-06-94
		WO 9414280 A	23-06-94
		WO 9414281 A	23-06-94
		WO 9414282 A	23-06-94

INTERNATIONAL SEARCH REPORT

i...ormation on patent family members

Intern: al Application No
PCT/EP 97/02108

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9414284 A		WO 9414283 A	23-06-94
		US 5559549 A	24-09-96
		US 5600364 A	04-02-97
		US 5659350 A	19-08-97
US 5144663 A	01-09-92	AU 599646 B	26-07-90
		AU 7157887 A	22-10-87
		DE 3751410 D	24-08-95
		DE 3751410 T	11-04-96
		EP 0243312 A	28-10-87
		EP 0626793 A	30-11-94
		ES 2076931 T	16-11-95
		JP 2610260 B	14-05-97
		JP 63023488 A	30-01-88
		JP 2520217 B	31-07-96
		JP 5244591 A	21-09-93

Form PCT/ISA/210 (patent family annex) (July 1992)

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 25/02

A1

(11) International Publication Number: WO 98/19431

(43) International Publication Date: 7 May 1998 (07.05.98)

(21) International Application Number: PCT/US97/19676

(22) International Filing Date: 27 October 1997 (27.10.97)

(30) Priority Data:
08/741,273 30 October 1996 (30.10.96) US

(71) Applicant: QUALCOMM INCORPORATED [US/US]; 6455 Lusk Boulevard, San Diego, CA 92121 (US).

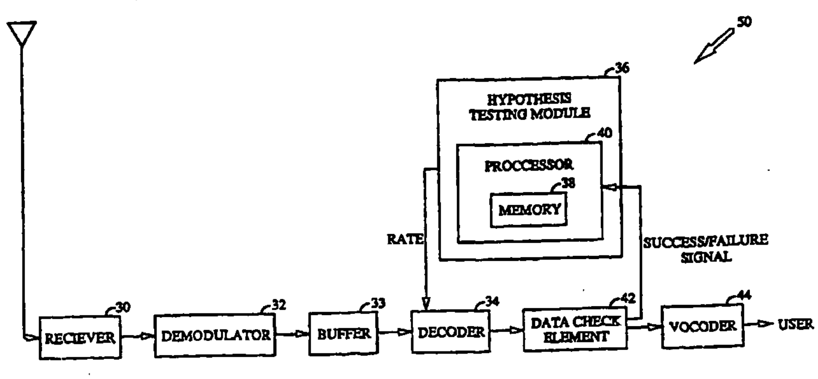
(72) Inventors: TIEDEMANN, Edward, G., Jr.; 4350 Bromfield Avenue, San Diego, CA 92122 (US). LIN, Yu-Chuan; 585 W. 63rd Avenue, Vancouver, British Columbia V6P 2G7 (CA).

(74) Agents: OGROD, Gregory, D. et al.; Qualcomm Incorporated, 6455 Lusk Boulevard, San Diego, CA 92121 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, 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, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, 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
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.

(54) Title: METHOD AND APPARATUS FOR DECODING VARIABLE RATE DATA



RECIEVING SYSTEM

(57) Abstract

A system and method for determining the data rate of a frame of data at a receiver (50) of a variable rate communications system. A vocoder at a transmitter encodes a frame of data at one of the rates of a predetermined set of rates. The data rate is dependent on the speech activity during the time frame of the data. The data frame is also formatted with overhead bits, including bits for error detection and detection. At the receiver (50), the data rate for the frame is determined based on hypothesis testing. Because the data rate is based on speech activity, a hypothesis test may be designed based on the statistics of speech activity. The received data frame is first decoded by a decoder (34) into information bits at the most probable rate as provided by the hypothesis testing module (36). Data check element (42) generates error metrics for the decoded information bits. If the error metrics indicate that the information bits are of good quality, then the information bits are presented to a vocoder (44) at the receiver to be processed for interface with the user. If the error metrics indicate that the information bits have not been properly decoded, then decoder (34) decodes the received data frame at the other rates of the set of rates until the actual data rate is determined.

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
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

METHOD AND APPARATUS FOR DECODING VARIABLE RATE DATA

BACKGROUND OF THE INVENTION

I. Field of the Invention

The present invention relates to digital communications. More particularly, the present invention relates to a novel and improved system and method for determining, at a receiver of a variable rate communication system, the rate at which data has been encoded for transmission.

II. Description of the Related Art

The use of code division multiple access (CDMA) modulation techniques is one of several techniques for facilitating communications in which a large number of system users are present. Although other techniques such as time division multiple access (TDMA), frequency division multiple access (FDMA), and AM modulation schemes such as amplitude companded single sideband (ACSSB) are known, CDMA has significant advantages over these other techniques. The use of CDMA techniques in a multiple access communication system is disclosed in U.S. Pat. No. 4,901,307, entitled "SPREAD SPECTRUM MULTIPLE ACCESS COMMUNICATION SYSTEM USING SATELLITE OR TERRESTRIAL REPEATERS," assigned to the assignee of the present invention and incorporated by reference herein.

CDMA systems often employ a variable rate vocoder to encode data so that the data rate can be varied from one data frame to another. An exemplary embodiment of a variable rate vocoder is described in U.S. Pat. No. 5,414,796, entitled "VARIABLE RATE VOCODER," assigned to the assignee of the present invention and incorporated by reference herein. The use of a variable rate communications channel reduces mutual interference by eliminating unnecessary transmissions when there is no useful speech to be transmitted. Algorithms are utilized within the vocoder for generating a varying number of information bits in each frame in accordance with variations in speech activity. For example, a vocoder with a set of four rates may produce 20 millisecond data frames containing 16, 40, 80, or 171 information bits, depending on the activity of the speaker. It is desired to transmit each data frame in a fixed amount of time by varying the transmission rate of communications.

Additional details on the formatting of the vocoder data into data frames are described in U.S. Pat. No. 5,511,073, entitled "METHOD AND APPARATUS FOR THE FORMATTING OF DATA FOR TRANSMISSION," assigned to the assignee of the present invention and herein incorporated by reference. The data frames may be further processed, spread spectrum modulated, and transmitted as described in U.S. Pat. No. 5,103,459, entitled "SYSTEM AND METHOD FOR GENERATING WAVEFORMS IN A CDMA CELLULAR TELEPHONE SYSTEM," assigned to the assignee of the present invention and incorporated by reference herein.

Variable rate systems can be developed which include explicit rate information. If the rate is included as part of a variable rate frame, then the rate is not recoverable until after the frame has already been properly decoded, at which point the rate has already been determined. Rather than including the rate in a variable rate frame, the rate could instead be sent in a non-variable rate portion of the frame. However, only a few bits are typically needed to represent the rate, and these bits cannot be efficiently encoded and interleaved in order to provide error protection for fading communications channels. Furthermore, the rate information is only available after some decoding delay and are subject to error.

Alternatively, variable rate systems can be developed which do not include explicit rate information. One technique for the receiver to determine the rate of a received data frame where the rate information is not explicitly included in the frame is described in copending U.S. Patent Application Serial No. 08/233,570, entitled "METHOD AND APPARATUS FOR DETERMINING DATA RATE OF TRANSMITTED VARIABLE RATE DATA IN A COMMUNICATIONS RECEIVER," filed April 26, 1994, assigned to the assignee of the present invention, and incorporated by reference. Another technique is described in copending U.S. Patent Application Serial No. 08/126,477, entitled "MULTIRATE SERIAL VITERBI DECODER FOR CODE DIVISION MULTIPLE ACCESS SYSTEM APPLICATIONS," filed Sept. 24, 1993, assigned to the assignee of the present invention, and incorporated by reference. According to these techniques, each received data frame is decoded at each of the possible rates. Error metrics, describing the quality of the decoded symbols for each frame decoded at each rate, are provided to a processor. The error metrics may include Cyclic Redundancy Check (CRC) results, Yamamoto Quality Metrics, and Symbol Error Rates. These error metrics are well-known in communications systems. The processor analyzes the error metrics and determines the most probable rate at which the incoming symbols were transmitted.

Decoding each received data frame at each possible data rate will eventually generate the desired decoded data. However, the search through all possible rates is not the most efficient use of processing resources in a receiver. Also, as higher transmission rates are used, power consumption for determining the transmission rate also increases. This is because there are more bits per frame to be processed. Furthermore, as technology evolves, variable rate systems may utilize larger sets of data rates for communicating information. The use of larger sets of rates will make the exhaustive decoding at all possible rates infeasible. In addition, the decoding delay will not be tolerable for some system applications. Consequently, a more efficient rate determination system is needed in a variable rate communications environment. These problems and deficiencies are clearly felt in the art and are solved by the present invention in the manner described below.

SUMMARY OF THE INVENTION

The present invention is a novel and improved system and method for determining the transmission rate of communications in a variable rate communications system. In a variable rate system, the data rate at which a data frame is encoded may be based on the speech activity during the time frame. Because the characteristics of speech are known, probability functions may be defined for the data rates which are dependent on the characteristics of speech. The probability functions may in addition be dependent on the measured statistics of the received data frames. Furthermore, hypothesis tests can be designed based on the probability functions to determine the most likely data rate of a received frame of data. These probability functions may be dependent on the selected service option. For example, the probability functions for data services will be different than for voice services.

At the receiver of the present invention, a processor causes a decoder to decode the received frame of data into information bits at the most probable rate as determined by the hypothesis test. The most probable rate may, for example, be the rate of the previous frame of data. The decoder also generates error metrics for the decoded information bits. The decoded bits and the error metrics are provided to a data check element which checks the decoded bits for correctness. If the error metrics indicate that the decoded information bits are of good quality, then the information bits are provided to a vocoder which further processes the data and provides speech to the user. Otherwise, a failure signal is presented to the processor. The processor then causes the decoder to decode the received frame of data at other data rates until the correct data rate is found.

BRIEF DESCRIPTION OF THE DRAWINGS

The features, objects, and advantages of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference characters identify correspondingly throughout and wherein:

FIG. 1 is a schematic overview of an exemplary CDMA cellular telephone system;

FIG. 2 is a block diagram of a variable rate receiving system with particular reference to the rate determination features of the present invention;

FIGS. 3 and 4 are flow charts illustrating two embodiments of the processing steps involved in rate determination wherein the hypothesis test designates the rate of the previous frame of data as the most probable rate for the current frame of data;

FIGS. 5 and 6 are flow charts illustrating two embodiments of the processing steps involved in rate determination wherein the hypothesis test is based on the a priori probability distribution of the data rates; and

FIGS. 7 and 8 are flow charts illustrating two embodiments of the processing steps involved in rate determination wherein the hypothesis test is based on the conditional probability distribution of the data rates.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

An exemplary cellular mobile telephone system in which the present invention is embodied is illustrated in FIG. 1. For purposes of example this system is described herein within the context of a CDMA cellular communications system. However, it should be understood that the invention is applicable to other types of communication systems such as personal communication systems (PCS), wireless local loop, private branch exchange (PBX) or other known systems. Furthermore systems utilizing other well known transmission modulation schemes such as TDMA and

FDMA as well as other spread spectrum systems may employ the present invention.

An exemplary cellular system in which the rate determination system of the present invention may be implemented is illustrated in FIG. 1. In FIG. 1, system controller and switch **10** typically include appropriate interface and processing hardware for providing system control information to the cell-sites. Controller **10** controls the routing of telephone calls from the public switched telephone network (PSTN) to the appropriate cell-site for transmission to the appropriate mobile unit. Controller **10** also controls the routing of calls from the mobile units via at least one cell-site to the PSTN. Controller **10** may direct calls between mobile users via the appropriate cell-site stations since such mobile units do not typically communicate directly with one another.

Controller **10** may be coupled to the cell-sites by various means such as dedicated telephone lines, optical fiber links or by radio frequency communications. In FIG. 1, two exemplary cell-sites, **12** and **14**, along with two exemplary mobile units, **16** and **18**, which include cellular telephones, are illustrated. Arrows **20a-20b** and **22a-22b** respectively define the possible communication links between cell-site **12** and mobile units **16** and **18**. Similarly, arrows **24a-24b** and arrows **26a-26b** respectively define the possible communication links between cell-site **14** and mobile units **18** and **16**.

The cellular system illustrated in FIG. 1 may employ a variable rate data channel for communications between cell-sites **12, 14** and mobile units **16, 18**. By example, a vocoder (not shown) may encode sampled voice information into symbols at four different rates according to the IS-95-A standard. The IS-95-A Mobile Station-Base Station Compatibility Standard for Dual Mode Wideband Spread Spectrum Cellular System has been provided by the telecommunications industry association (TIA) for CDMA communications. According to IS-95-A, speech is encoded at approximately 8,550 bits per second (bps), 4,000 bps, 2,000 bps, and 800 bps based on voice activity during a 20 millisecond (ms) frame of data. Each frame of vocoder data is then formatted with overhead bits as 9,600 bps, 4,800 bps, 2,400 bps, and 1,200 bps data frames for transmission. The 9,600 bps frame is referred to as a full rate frame; the 4,800 bps data frame is referred to as a half rate frame; a 2,400 bps data frame is referred to as a quarter rate frame; and a 1,200 bps data frame is referred to as an eighth rate frame. Although this example describes a set of four data rates of the IS-95-A standard, it should be recognized that the present invention is equally applicable in systems utilizing different transmission rates and/or a different number of variable rates.

By encoding each frame of data based on speech activity, data compression is achievable without impacting the quality of the reconstructed speech. Since speech inherently contains periods of silence, i.e. pauses, the amount of data used to represent these periods can be reduced. Variable rate vocoding most effectively exploits this fact by reducing the data rate for these periods of silence. In a system with a set of four rates as described above, periods of active speech will generally be encoded at full rate, while periods of silence will generally be encoded at eighth rate. Most frames (about 80-90%) are encoded at full or eighth rate. Transitions between active speech and periods of silence will typically be encoded at half or quarter rate. An exemplary encoding technique which compresses data based on speech activity is described in U.S. Pat. No. 5,511,073 mentioned above.

The data frames are also formatted with overhead bits, which generally will include additional bits for error correction and detection, such as Cyclic Redundancy Check (CRC) bits. The CRC bits can be used by the decoder to determine whether or not a frame of data has been received correctly. CRC codes are produced by dividing the data block by a predetermined binary polynomial as is described in detail in IS-95-A.

In a preferred embodiment, each frame of symbol data is interleaved by an interleaver, preferably on a bit level basis, to increase time diversity for purposes of error detection. The formatted data frames undergo further processing, which include modulation, frequency upconversion to the radio frequency (RF) and amplification of the signals of data frames, before transmission.

When signals of the variable rate data frames are received by a receiver, the receiver must determine the rate of transmission in order to properly decode the signals. However, the rate of the received frame is not known by the mobile station a priori. Therefore, some other method of ascertaining the rate is necessary.

The present invention accomplishes rate determination through the use of hypothesis testing. Hypothesis tests are designed based on the probability distribution of the data rates of the frames of speech. Although the data rate of each received frame is not known a priori, the probability of receiving a frame at a given rate can be determined. As mentioned above, a variable rate vocoder encodes each frame of speech at one of a set of predetermined rates based on the speech activity during the time frame.

Since the characteristics of speech activity can be modeled, probabilistic functions of the data rates which depend on speech activity can be derived from the model. Hypothesis tests can then be designed based on the probabilistic functions of data rates to determine the most likely data rate for each received frame of data.

The use of hypothesis testing for rate determination in a variable rate receiving system may be better appreciated by referring to FIG. 2. In a CDMA environment, for example, the receiving system **50** of FIG. 2 may be implemented in either a mobile unit or a cell site in order to determine the data rate of received signals. The present invention offers particular advantages because it avoids the exhaustive decoding at all rates. By choosing a hypothesis and checking the hypothesis for correctness, the average amount of processing for each received frame is reduced. This is especially important in the mobile unit because reduced processing, and thereby power consumption, in the decoding process can extend battery life in the receiver.

The variable rate receiving system **50** illustrated in FIG. 2 includes receiver **30** for collecting transmitted signals, including the data signal of interest. Receiver **30** amplifies and frequency downconverts the received signals from the RF frequency band to the intermediate frequency (IF) band.

The IF signals are presented to demodulator **32**. The design and implementation of demodulator **32** are described in detail in U.S. Pat. No. 5,490,165, entitled "DEMODULATION ELEMENT ASSIGNMENT IN A SYSTEM CAPABLE OF RECEIVING MULTIPLE SIGNALS," issued Feb. 6, 1996, and assigned to the assignee of the present invention, the disclosure of which is incorporated by reference herein. Demodulator **32** demodulates the IF signal to produce a data signal consisting of the symbols of one frame of data. Demodulator **32** generates the data signal by despreading and correlating the IF signal addressed to the receiver. The demodulated data signal is then fed to buffer **33**. Buffer **33** stores the demodulated data signal, or the received symbols, until it is properly decoded. Buffer **33** may also be the deinterleaver if the data frame had been interleaved for transmission. Buffer **33** provides the demodulated symbol data to decoder **34**.

Hypothesis testing module **36** implements the hypothesis test for determining the data rate of a received frame of data. Hypothesis testing module **36** comprises processor **40**, which includes memory **38**. The information needed in hypothesis testing such as the decoded rates from the previous frames and the probabilities are stored in memory **38**. For each data frame received, processor **40** determines the most probable rate based on the information stored in memory **38**. Processor **40** then presents the most probable data rate to decoder **34** which decodes the data signal at this most probable rate to produce decoded bits.

In the exemplary embodiment, decoder **34** is a trellis decoder capable of decoding data of varying rates, such as a Viterbi decoder. The design and implementation of a multirate Viterbi decoder which exhaustively decodes a received signal at all rates of a set of rates is described in the aforementioned U.S. Patent Applications 08/233,570 and 08/126,477. It will be understood by one skilled in the art that the multirate Viterbi decoder may be modified to decode at a selected rate. This may be accomplished by having the Viterbi decoder receive a rate indicator input, in response to which the decoder decodes the data signal according to the rate indicator. Thus, the modified Viterbi decoder may decode a received data frame based on a rate indicator supplied by processor **40** of hypothesis testing module **36**.

Decoder **34** generates information data bits and error metrics characterizing the information bits. The error metrics include the previously described CRC bits, which were added into the data frames as overhead bits. Decoder **34** may also generate other error metrics, such as the Yamamoto Quality Metric and the Symbol Error Rate (SER). The Yamamoto metric is determined by comparing the differences in the metrics of remerging paths in each step of the Viterbi decoding with a threshold and labeling a path as unreliable if the metric difference is less than a quality threshold. If the final path selected by the Viterbi decoder has been labeled as unreliable at any step, the decoder output is labeled as unreliable. Otherwise, it is labeled as reliable. The Symbol Error Rate is determined by taking the decoded bits, re-encoding these bits to provided re-encoded symbols, and comparing these re-encoded symbols against the received symbols which are stored in buffer **33**. The SER is a measure of the mismatching between the re-encoded symbols and the received symbols. The decoded information bits and the error metrics are provided to data check element **42**, which determines if the information bits have been correctly decoded.

In a preferred embodiment, data check element **42** first checks the CRC bits. If the CRC check fails, then data check element **42** provides a signal indicative of the failure to processor **40**. If the CRC check passes, then data check element **42** determines if the re-encoded SER is below a certain threshold. If the SER is above the threshold, then a signal indicative of failure is provided to processor **40**. Otherwise, the data rate provided by hypothesis testing module **36** is determined to be correct, and a success signal is provided to processor **40**, whereupon no further decoding is performed on the data frame. The properly decoded data signal is presented to variable rate vocoder **44**.

When processor **40** receives a failure signal indicating that data symbols have not been properly decoded into information bits, processor **40** will determine at least one other data rate from the set of data rates at which to decode the data symbols. Processor **40** provides the rate information to decoder **34**, which decodes the data symbols at the rate provided. For each data rate at which the data signal is decoded, data check element **42** will determine the quality of the decoded information bits. Upon determination by data check element **42** that the correct data rate has been found, a signal of decoded information bits is provided to variable rate vocoder **44**. Vocoder **44** will then process the information bits for interface with the user.

Hypothesis testing module **36** may implement any of a number of hypothesis tests for determining the data rate of a received frame of data. For example the hypothesis test may be based on known statistics of speech activity. It is known that for a set of four rates using 20 ms frames, a full rate frame will usually be followed by another full rate frame, while an eighth-rate frame will usually be followed by another eighth rate frame. Further, it is also known that most frames will either be full or eighth rate rather than half or quarter rate, because the periods of speech and silence do not occur in 20 ms bursts. Based on these characteristics, the hypothesis test may designate the rate of the previous frame of data as the most probable rate for the currently received frame of data.

In an exemplary implementation, the rate of the previous frame of data is stored in memory **38** of hypothesis testing module **36**. When a data frame is received, processor **40** of hypothesis testing module **36** obtains the rate of the previous frame from memory **38** and presents it to decoder **34**. Decoder **34** decodes the received data frame at the rate of the previous frame to produce information bits. Decoder **34** also generates error metrics which are then presented to data check element **42** along with the information bits. If data check element **42** determines from the error metrics that the decoded bits are of good quality, then the information bits are presented to vocoder **44**. Otherwise, a failure indication is sent from data check element **42** to processor **40**. Processor **40** may then have decoder **34** exhaustively decode the data frame at all other rates before determining the data rate. A flow chart illustrating some of the steps involved in rate determination as described in the embodiment above is shown in FIG. 3.

Alternatively, processor **40** may have decoder **34** sequentially decode the data frame according to a ranking from the next most likely rate to the least likely rate. The ranking may be determined in a number of ways, such as according to the probability distributions described below. For each decoding, error metrics are generated by decoder **34** and checked by data check element **42** for correctness. When correctly decoded, the decoded frame is passed on to vocoder **44**. A flow chart illustrating some of the processing steps of this embodiment is shown in FIG. 4.

Another implementation of hypothesis testing module **36** is based upon the a priori probability distribution of data rates. For a set of four rates, the a priori probability distribution (P) of the data rates may be defined as:

$$P = \mathrm{Prob}\{R_t\}, \tag{3}$$

where R_t refers to the full, half, quarter, or eighth rate at time t. The likelihood of receiving a frame at each of the different data rates of a set of rates are maintained in memory **38** of processor **40**. Generally, the probability distribution of the data rates are determined based on the theoretical statistics or the empirical statistics of speech activity. The likelihood of receiving a frame at the different rates are then permanently stored in memory **38** for determining the rate of every received frame of data. In a more sophisticated embodiment, the likelihood of the rates stored in memory **38** may be updated based on the actual statistics of the received frames of data.

For each new frame of data received, processor **40** obtains the most probable rate from memory **38** and presents the most probable rate to decoder **34**. Decoder **34** decodes the data signal at this most probable data rate and presents the decoded data to data check element **42**. Error metrics, including the CRC, are also generated by decoder **34** and presented to data check element **42**. Other error metrics may also be generated for checking by data check element **42**. If the error metrics indicate that the decoded bits are of good quality, then the information bits are presented to vocoder **44**. Otherwise, a failure indication is sent from data check element **42** to processor **40**. Then, processor **40** obtains the second most likely data rate from memory **38** and presents it to decoder **34**, and the process of decoding and error checking is continued until the correct data rate is found. A flow chart of the processing steps of this embodiment is illustrated in FIG. 5. Alternatively, upon receipt of a failure signal by processor **40**, processor **40** may cause decoder **34** to exhaustively decode the data frame at each of the other data rates of the set of rates, and error metrics are checked for each decoding in order to determine the actual rate of transmission. A flow chart of the processing steps of this embodiment is illustrated in FIG. 6.

Instead of designing the hypothesis test based on the simple probability distribution of the data rates, conditional probabilities may be used to improve on the accuracy of the rate determination. For example, the probability of receiving a data frame at a given rate may be defined to be conditioned on the actual rates of the previous frames of data. Conditional probabilities based on the previous rates work well because transition characteristics of the data signals are well known. For example, if the rate two frames ago was eighth rate and the rate for the previous frame was half rate, then the most likely rate for the current frame is full rate, because the transition to half rate indicates the onset of active speech. Conversely, if the rate two frames ago was full rate and the rate for the previous frame was quarter rate, then the most likely rate for the present frame might be eighth rate, because the rate transition indicates the onset of silence.

The probability distribution of the data rates conditioned on the rates of the previous n frames of data may be defined as:

$$P = \mathrm{Prob}\{\, R_t \mid R_{t-1}, R_{t-2}, \ldots, R_{t-n}\} \qquad (4)$$

where R_t again refers to the rate at time t, and $R_{t-1}, R_{t-2}, \ldots, R_{t-n}$ refers to rate(s) of the previous n frame(s) of data, for $n \geq 1$. The likelihood of receiving a frame at each of the different data rates of a set of rates conditioned on the previous n actual rates are stored in memory **38** of processor **40**. In addition, the actual data rates of the previous n frames of data are maintained by processor **40**, and may be stored in memory **38** as the rates are determined.

For each received frame of data, processor **40** will determine the most probable data rate conditioned on the previous n actual data rates and present it to decoder **34**. Decoder **34** will decode the frame at this most probable data rate and present the decoded bits to data check element **42**. In addition, error metrics are generated by decoder **34** and presented to data check element **42**. If the error metrics indicate that the decoded bits are of good quality, then the information bits are presented to vocoder **44**. Also, processor **40** is informed of the rate decision so that it can maintain the history of chosen rates. That is, processor **40** is supplied R_t so that it can be used in determining $\mathrm{Prob}\{\, R_t \mid R_{t-1}, R_{t-2}, \ldots, R_{t-n}\}$ for the next frame. If error metrics indicate an unsuccessful decoding, then a failure indication signal is sent from data check element **42** to processor **40**, and processor **40** determines the second most probable data rate conditioned on the previous n actual data rates to decode the data frame. As in the simple probabilities case, the process of decoding and error checking is continued until the correct data rate is found. Some of the processing steps of this embodiment are illustrated in a flow chart in FIG. 7. Also as in the simple probabilities case, after a failed decoding at the most likely rate, decoder **34** may exhaustively decode the data frame at all of the other data rates and have error metrics checked for all decoding in order to determine the data rate. Some of the processing steps of this embodiment are illustrated in a flow chart in FIG. 8.

It should be understood that the conditional probability distribution of the data rates may depend on statistics other than the actual rates of the previous frames of data. For example, the probability distribution may be conditioned on one or more frame quality measurements. The probability distribution is then defined to be:

$$P = \mathrm{Prob}\{R_t \mid X_1, X_2, \ldots, X_k\}, \tag{5}$$

where R_t is the rate at time t, and $X_1, X_2, \ldots X_k$ are one or more frame quality measurements. The k frame quality measurements may be measurements performed on the current frame of data, or measurements performed on previous frame(s) of data, or a combination of both. An example of a frame quality measurement is the SER error metric mentioned above. Thus, the probability of receiving a frame at a given rate is conditioned on the SER obtained from the previous decoding if a previous decoding had been performed.

The conditional probability distribution may also depend on a combination of the actual rates of the previous frames of data and the frame quality measurements. In this case, the probability distribution of the data rates is defined as:

$$Pt = \mathrm{Prob}\{ R_t \mid R_{t-1}, R_{t-2}, \ldots, R_{t-n}, X_1, X_2, \ldots, X_k \}, \tag{6}$$

where R_t is the rate at time t, $R_{t-1}, R_{t-2}, \ldots, R_{t-n}$ are the rates of the previous frames of data and $X_1, X_2, \ldots, X_k$ are the frame quality measurements.

In the cases where the probability distribution is based on frame quality measurements, the frame quality measurements should be maintained in processor **40** of hypothesis testing module **36**. As can be seen from the above description, the hypothesized frame rate may be conditioned on a number of different statistics, and the rates of the previous frames and the frame quality measurements are examples of these statistics. For each data frame received, processor **40** uses the statistics to determine the rate at which to decode the frame.

A further refinement to the determination of the rate at which to decode a received frame of data considers the processing costs of decoding the frame at the various rates in conjunction with hypothesis testing. In this embodiment, an optimum test sequence of the rates is established based on both the probability distribution of the data rates and the cost of decoding at each of the data rates. The optimum test sequence is maintained by processor **40**, which causes decoder **34** to sequentially decode a received frame of data according to the optimum sequence until the correct rate is found. The optimum test sequence is established to minimize the total expected cost of the rate search. Denoting P_i to be the probability that the rate search will stop at test T_i, and C_i to be the cost for conducting test T_i, the total expected cost of the rate search using test sequence $T_1, T_2, \ldots, T_M$, where M is the number of possible rates in the system and $1 \leq i \leq M$, can be modeled as:

$$C_{total} = C_1 * P_1 + (C_1 + C_2)* P_2 + \ldots (C_1 + C_2 +\ldots+ C_M)* P_M. \quad (7)$$

The optimum test sequence is found by minimizing the total expected cost C_{total}.

In Equation (7), the cost C_i for conducting test T_i will generally be the processing power required for decoding a frame at the rate specified by test T_i. The cost may be assigned to be proportional to the frame rate specified by the test T_i because the computational complexity of decoder **34** is in general approximately proportional to the number of bits per frame. The probabilities P_i may be assigned by the unconditioned a priori probability distribution of data rates as defined by Equation (3), or any of the conditional probability distributions defined by Equations (4), (5), or (6) above.

In a variable rate communications system where data frames are transmitted at 9,600 bps, 4,800 bps, 2,400 bps, and 1,200 bps, the following example illustrates the formulation of the optimum test sequence for rate determination of a received frame. The costs to decode the 9,600 bps, 4,800 bps, 2,400 bps, and 1,200 bps frames are assumed to be 9.6, 4.8, 2.4, and 1.2, respectively. Further, the probability of receiving a frame at each of the four rates is assumed to be the unconditioned a priori probabilities having the following values:

Prob(9,600 bps) = 0.291, (8)

Prob(4,800 bps) = 0.039, (9)

Prob(2,400 bps) = 0.072, and (10)

Prob(1,200 bps) = 0.598. (11)

The probabilities given in Equations (8)-(11) are derived from steady state empirical data.

A listing of all possible test sequences for rate determination in the system transmitting frames at 9,600, 4,800, 2,400, and 1,200 bps is shown in Table I below. In Table I, column 1 lists all possible test sequences T_1, T_2, T_3, T_4, where $T_i = 1$ refers to the test of decoding at 9,600 bps, $T_i = 1/2$ refers to the test of decoding at 4,800 bps, $T_i = 1/4$ refers to the test of decoding at 2,400 bps, and $T_i = 1/8$ refers to the test of decoding at 1,200 bps. Columns 2 and 3 list the probability P_1 and the cost C_1 of performing the test T_1, columns 4 and 5 list the probability P_2 and the cost C_2 of performing the test T_2, columns 6 and 7 list the probability P_3 and the cost C_3 of performing the test T_3, and columns 8 and 9 list the probability P_4 and the cost C_4 of performing the test T_4. The total cost C_{total} of performing the test sequence T_1, T_2, T_3, T_4 is listed in column 10.

T_1, T_2, T_3, T_4	P_1	C_1	P_2	C_2	P_3	C_3	P_4	C_4	C_{total}
1, 1/2, 1/4, 1/8	0.291	9.6	0.039	4.8	0.072	2.4	0.598	1.2	15.33
1, 1/2, 1/8, 1/4	0.291	9.6	0.039	4.8	0.598	1.2	0.072	2.4	13.98
1, 1/4, 1/2, 1/8	0.291	9.6	0.072	2.4	0.039	4.8	0.598	1.2	15.08
1, 1/4, 1/8, 1/2	0.291	9.6	0.072	2.4	0.598	1.2	0.039	4.8	12.25
1, 1/8, 1/2, 1/4	0.291	9.6	0.598	1.2	0.039	4.8	0.072	2.4	11.16
1, 1/8, 1/4, 1/2	0.291	9.6	0.598	1.2	0.072	2.4	0.039	4.8	10.90
1/2, 1, 1/4, 1/8	0.039	4.8	0.291	9.6	0.072	2.4	0.598	1.2	16.35
1/2, 1, 1/8, 1/4	0.039	4.8	0.291	9.6	0.598	1.2	0.072	2.4	15.00
1/2, 1/4, 1, 1/8	0.039	4.8	0.072	2.4	0.291	9.6	0.598	1.2	16.36
1/2, 1/4, 1/8, 1	0.039	4.8	0.072	2.4	0.598	1.2	0.291	9.6	10.97
1/2, 1/8, 1, 1/4	0.039	4.8	0.598	1.2	0.291	9.6	0.072	2.4	9.61
1/2, 1/8, 1/4, 1	0.039	4.8	0.598	1.2	0.072	2.4	0.291	9.6	9.62
1/4, 1, 1/2, 1/8	0.072	2.4	0.291	9.6	0.039	4.8	0.598	1.2	15.08
1/4, 1, 1/8, 1/2	0.072	2.4	0.291	9.6	0.598	1.2	0.039	4.8	12.26
1/4, 1/2, 1, 1/8	0.072	2.4	0.039	4.8	0.291	9.6	0.598	1.2	16.11
1/4, 1/2, 1/8, 1	0.072	2.4	0.039	4.8	0.598	1.2	0.291	9.6	10.71
1/4, 1/8, 1/2, 1	0.072	2.4	0.598	1.2	0.039	4.8	0.291	9.6	7.89
1/4, 1/8, 1, 1/2	0.072	2.4	0.598	1.2	0.291	9.6	0.039	4.8	6.87
1/8, 1, 1/4, 1/2	0.598	1.2	0.291	9.6	0.072	2.4	0.039	4.8	5.51
1/8, 1, 1/2, 1/4	0.598	1.2	0.291	9.6	0.039	4.8	0.072	2.4	5.76
1/8, 1/2, 1, 1/4	0.598	1.2	0.039	4.8	0.291	9.6	0.072	2.4	6.79
1/8, 1/2, 1/4, 1	0.598	1.2	0.039	4.8	0.072	2.4	0.291	9.6	6.79
1/8, 1/4, 1/2, 1	0.598	1.2	0.072	2.4	0.039	4.8	0.291	9.6	6.54
1/8, 1/4, 1, 1/2	0.598	1.2	0.072	2.4	0.291	9.6	0.039	4.8	5.52

Table I

As shown in Table I, the optimum test sequence is the sequence 1/8, 1, 1/4, 1/2 shown in the 19th row. This test sequence offers the lowest total expected cost of processing. Therefore, the rate determination system would decode a received frame of data at 1,200 bps first. If the decoding at 1.200 bps is not successful, then the frame would be decoded sequentially at 9,600 bps, 2,400 bps, and 4,800 bps until the correct rate is found. In a preferred embodiment, the optimum test sequence is maintained by processor **40** of hypothesis testing module **36**. For each frame of data received, processor **40** causes decoder **34** to decode the frame sequentially according to the optimum test sequence, with each decoding checked by data check element **42**, until the correct data rate is found. Processing resources are efficiently utilized in this rate determination system because the decoding is performed sequentially according to an optimum search sequence.

Based on the embodiments described above, it will be understood by one skilled in the art that the present invention is applicable to all systems in which data has been encoded according to a variable rate scheme and the data must be decoded in order to determine the rate. Even more generally, the invention is applicable to all systems in which the encoded data E is a function of the data D and some key k, and there exists some information in D or E which permits the verification of the correct D by the receiver. The sequence k may be time varying. The encoded data is represented as:

$$E = f(D,k), \tag{1}$$

where k is from a small set K of keys and where some probability function exists on the set of keys. The inverse of the encoding, or the decoding, can be represented as:

$$D = f^{-1}(E,k), \tag{2}$$

where k is chosen so that D is correct.

As an example, assume that D is data composed of fixed-length sequence D1 and fixed length sequence D2 so that D = D1, D2. Sequence D2 is the Cyclic Redundancy Code (CRC) of D1, so that D2 = fcrc(D1). Assume also that the encoding function, f(D,k), is an exclusive-OR of a fixed-length D with the fixed length sequence k. Then, the decoding, $f^{-1}(E,k)$, would be the exclusive-OR of E with the correct k. The correct k is verified by checking whether D2 = fcrc(D1). The correct k can be found by decoding all possible k's in K and then determining whether the CRC check passes. Alternatively, it can be done by sequentially decoding using one k at a time, with no further decoding once the "correct" k is found. According to the present invention, the order of sequential decoding is to be determined by hypothesis testing. A number of hypothesis tests, including the tests described above, may be utilized. The order of sequential decoding may in addition depend on the cost of processing, as described above. The use of hypothesis testing and/or cost functions in formulating a test sequence for rate determination reduces the average amount of processing as fewer k's will have to be tried.

The previous description of the preferred embodiments is provided to enable any person skilled in the art to make or use the present invention. The various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without the use of the inventive faculty. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

CLAIMS

1. In a variable rate communications system, a sub-system for determining, at a receiver, the data rate of a received data frame, comprising:

a processor for generating a signal indicating the most likely rate of said received data frame in accordance with a predetermined hypothesis test; and

a decoder for receiving said most likely rate signal and for decoding said received data frame into a decoded frame of bits at said most likely rate.

2. The rate determination sub-system of claim 1 wherein said most likely rate is the rate of the previous data frame.

3. The rate determination sub-system of claim 1 wherein said hypothesis test is based on an a priori probability distribution of data rates.

4. The rate determination sub-system of claim 1 wherein said hypothesis test is based on a conditional probability distribution of data rates conditioned on the rate of at least one previous data frame.

5. The rate determination sub-system of claim 1 wherein said hypothesis test is based on a conditional probability distribution of data rates conditioned on at least one frame quality measurement.

6. The rate determination sub-system of claim 1 further comprising a data check element for receiving said decoded bits, generating error metrics characterizing said decoded bits, and generating a quality indication based on said error metrics for said decoded bits.

7. The rate determination sub-system of claim 6,

further comprising a vocoder for receiving said decoded bits and processing said decoded bits to provide speech to an user upon generation of a positive indication of said quality; and

wherein upon generation of a negative indication of said quality, said processor further causes said decoder to perform additional decoding of said received data frame in accordance with at least one rate other than said most likely rate.

8. The rate determination sub-system of claim 7,

wherein said additional decoding is performed sequentially in accordance with a predetermined test sequence of data rates;

wherein said data check element generates error metrics for each said additional decoding and generates a quality indication based on said error metrics for each said additional decoding; and

wherein said additional decoding terminates upon generation of a positive indication of said quality.

9. The rate determination sub-system of claim 7,

wherein said additional decoding comprises exhaustive decoding of said received data frame at all rates of a rate set except said most likely rate; and

wherein said data check element generates error metrics for each said additional decoding and determines the rate of said received data frame in accordance with said error metrics.

10. The rate determination sub-system of claim 6 wherein said error metrics include a Cyclic Redundancy Check result.

11. The rate determination sub-system of claim 6 wherein said error metrics include a Symbol Error Rate metric.

12. The rate determination sub-system of claim 6 wherein said error metrics include a Yamamoto quality metric.

13. The rate determination sub-system of claim 1 wherein said processor comprises a memory for storing said most likely rate.

14. The rate determination sub-system of claim 1 wherein said decoder is a Viterbi decoder.

15. In a variable rate communications system, a sub-system for determining, at a receiver, the data rate of a received data frame, comprising:

a processor for generating a test sequence of data rates for determining the rate of a received data frame, said test sequence being generated in accordance with a predetermined hypothesis test;

a decoder for decoding said received data frame sequentially according to said test sequence and generating a decoded frame of bits for each rate at which said received data frame is decoded;

a data check element for generating error metrics characterizing said decoded bits and for generating a quality indication based on said error metrics for each rate at which said received data frame is decoded; and

wherein no further decoding is performed upon generation of a positive indication of said quality.

16. The rate determination sub-system of claim 15 wherein said hypothesis test is based on an a priori probability distribution of data rates.

17. The rate determination sub-system of claim 15 wherein said hypothesis test is based on a conditional probability distribution of data rates conditioned on the rate of at least one previous data frame.

18. The rate determination sub-system of claim 15 wherein said hypothesis test is based on a conditional probability distribution of data rates conditioned on at least one frame quality measurement.

19. The rate determination sub-system of claim 16 wherein said test sequence is generated further in accordance with the cost of decoding said received data frame at each of said data rates.

20. The rate determination sub-system of claim 17 wherein said test sequence is generated further in accordance with the cost of decoding said received data frame at each of said data rates.

21. The rate determination sub-system of claim 18 wherein said test sequence is generated further in accordance with the cost of decoding said received data frame at each of said data rates.

22. The rate determination sub-system of claim 15 further comprising a vocoder for receiving said decoded bits and processing said decoded bits to provide speech to an user upon generation of a positive indication of said quality.

23. The rate determination sub-system of claim 15 wherein said error metrics include a Cyclic Redundancy Check result.

24. The rate determination sub-system of claim 15 wherein said error metrics include a Symbol Error Rate metric.

25. The rate determination sub-system of claim 15 wherein said error metrics include a Yamamoto quality metric.

26. The rate determination sub-system of claim 15 wherein said processor comprises a memory for storing said test sequence of data rates.

27. The rate determination sub-system of claim 15 wherein said decoder is a Viterbi decoder.

28. A method for determining the rate of a received data frame in a variable rate communications system, comprising the steps of:

receiving a wide-band signal;

demodulating said wide-band signal to produce a data signal, wherein said data signal has been transmitted at one of a set of possible transmission rates;

generating a test sequence of data rates for determining the rate of said data signal, said test sequence being generated in accordance with a predetermined hypothesis test;

decoding said data signal sequentially according to said test sequence to generate a decoded frame of bits for each rate at which said data signal is decoded;

generating error metrics characterizing said decoded frame of bits for each rate at which said data signal is decoded;

generating a quality indication based on said error metrics for each rate at which said data signal is decoded; and

upon generation of a positive indication of said quality, providing said decoded frame of bits to a vocoder which processes said decoded bits to provide speech to an user.

29. The method of claim 28 wherein said hypothesis test is based on an a priori probability distribution of data rates.

30. The method of claim 28 wherein said hypothesis test is based on a conditional probability distribution of data rates conditioned on the rate of at least one previous data frame.

31. The method of claim 28 wherein said hypothesis test is based on a conditional probability distribution of data rates conditioned on at least one frame quality measurement.

32. The method of claim 29 wherein said test sequence is generated further in accordance with the cost of decoding said received data frame at each of said data rates.

33. The method of claim 30 wherein said test sequence is generated further in accordance with the cost of decoding said received data frame at each of said data rates.

34. The method of claim 31 wherein said test sequence is generated further in accordance with the cost of decoding said received data frame at each of said data rates.

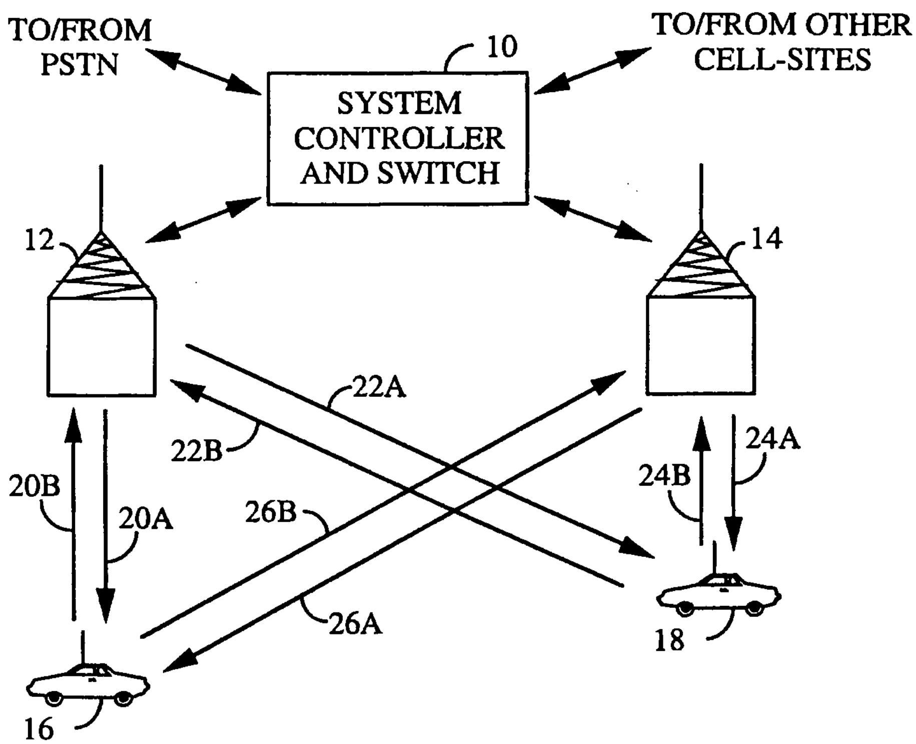
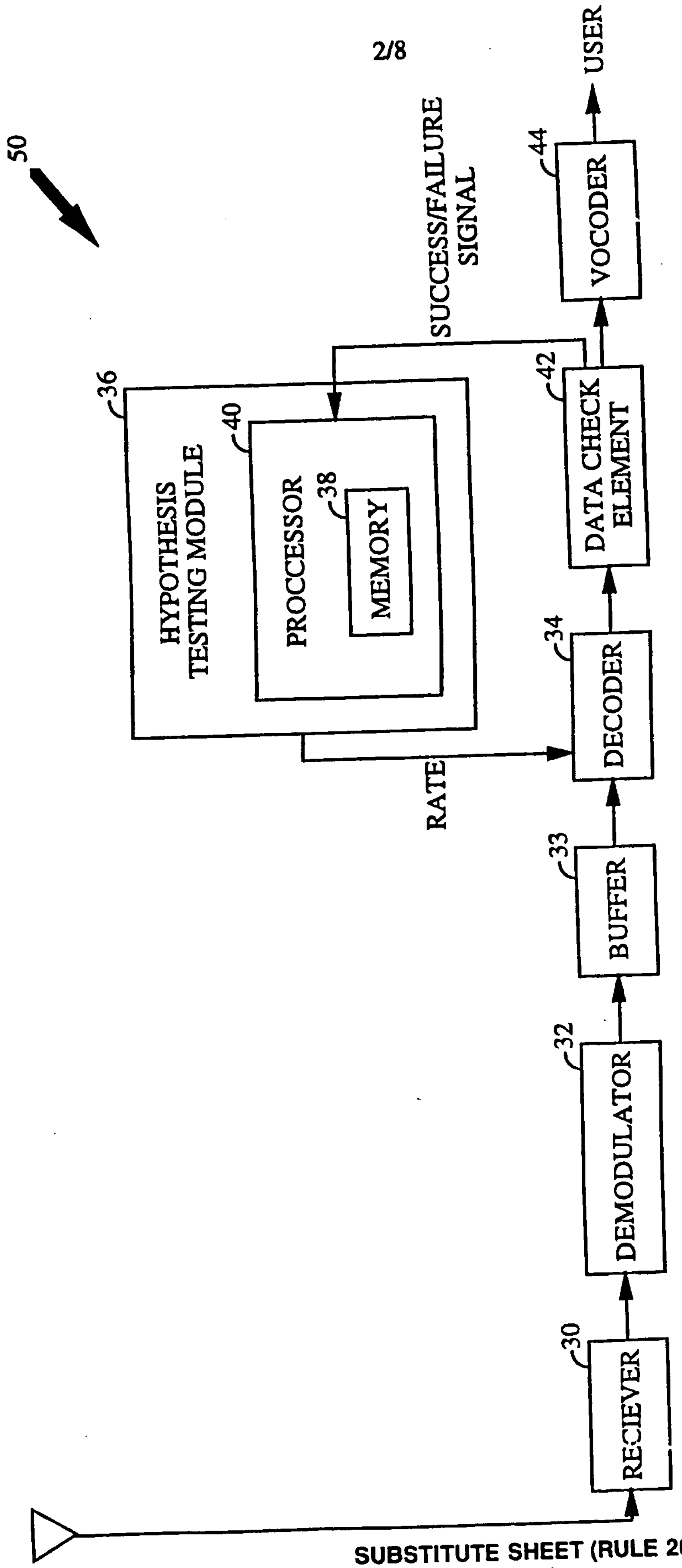

TO/FROM
PSTN
10
TO/FROM OTHER
CELL-SITES
SYSTEM
CONTROLLER
AND SWITCH
12
14
22A
22B
20B
20A
26B
24A
24B
26A
18
16

FIG. 1



RECIEVING SYSTEM

FIG. 2

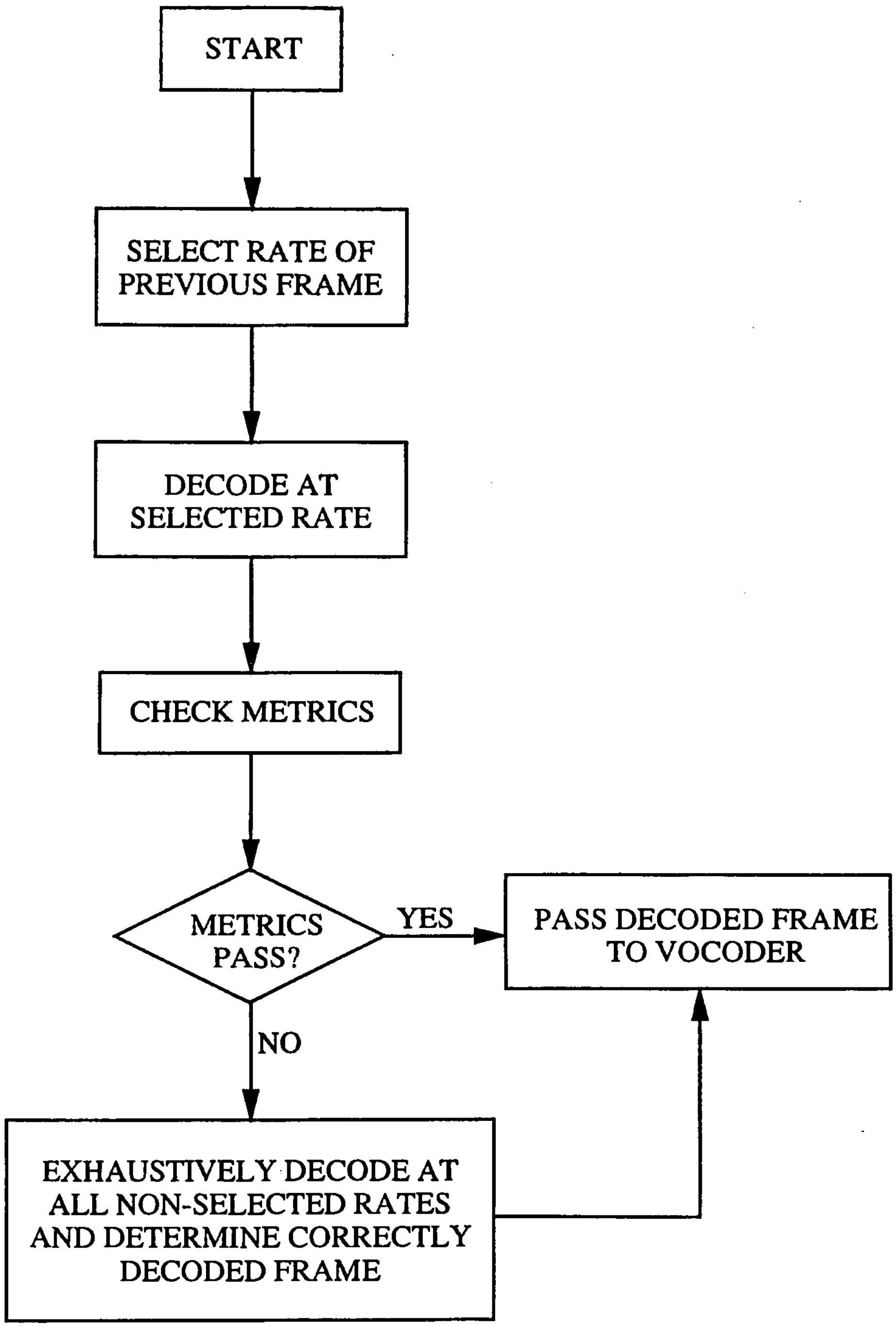

START
SELECT RATE OF PREVIOUS FRAME
DECODE AT SELECTED RATE
CHECK METRICS
METRICS PASS?
YES
PASS DECODED FRAME TO VOCODER
NO
EXHAUSTIVELY DECODE AT ALL NON-SELECTED RATES AND DETERMINE CORRECTLY DECODED FRAME

FIG. 3

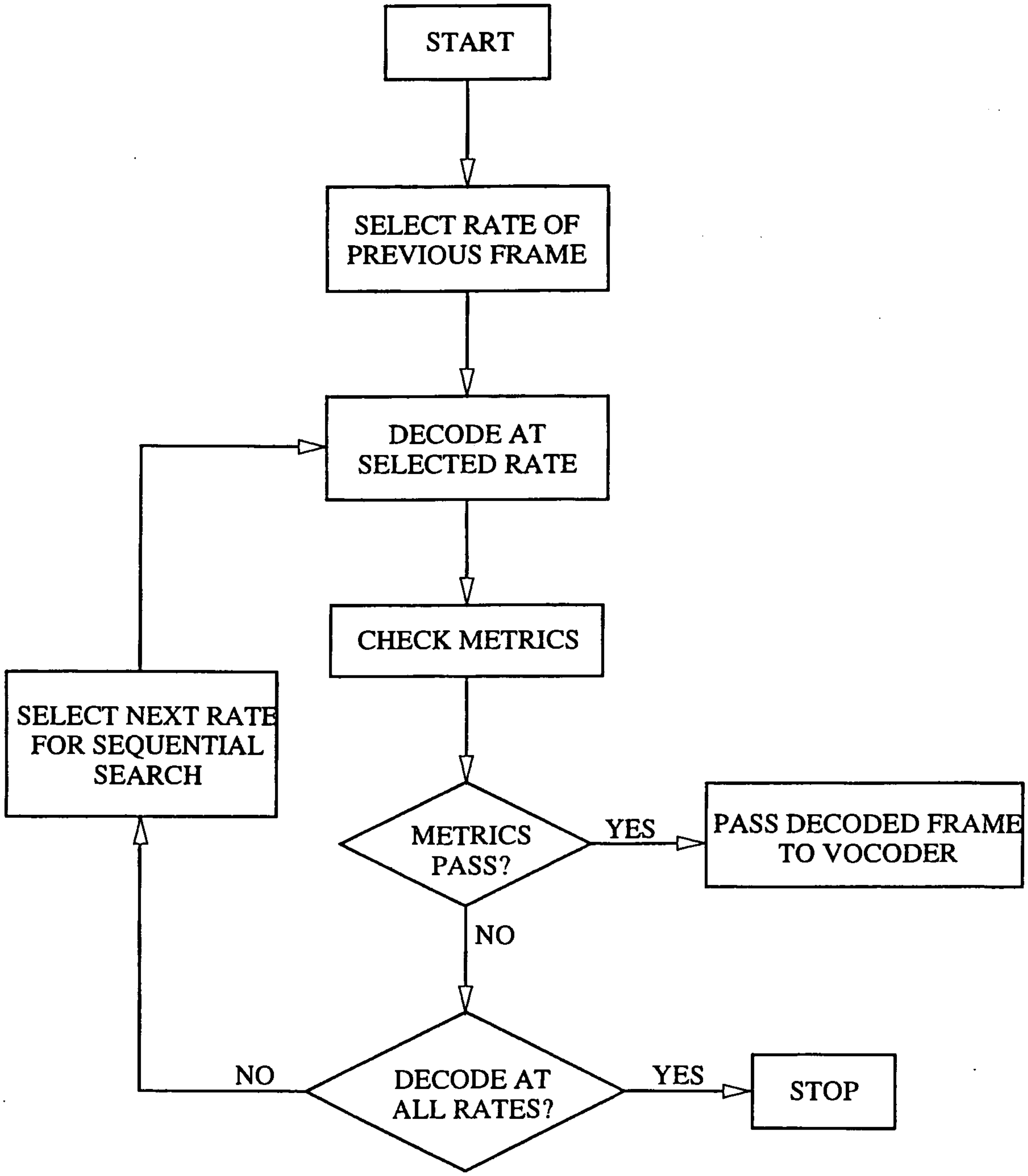

START
SELECT RATE OF PREVIOUS FRAME
DECODE AT SELECTED RATE
CHECK METRICS
SELECT NEXT RATE FOR SEQUENTIAL SEARCH
METRICS PASS?
YES
PASS DECODED FRAME TO VOCODER
NO
DECODE AT ALL RATES?
NO
YES
STOP

FIG. 4

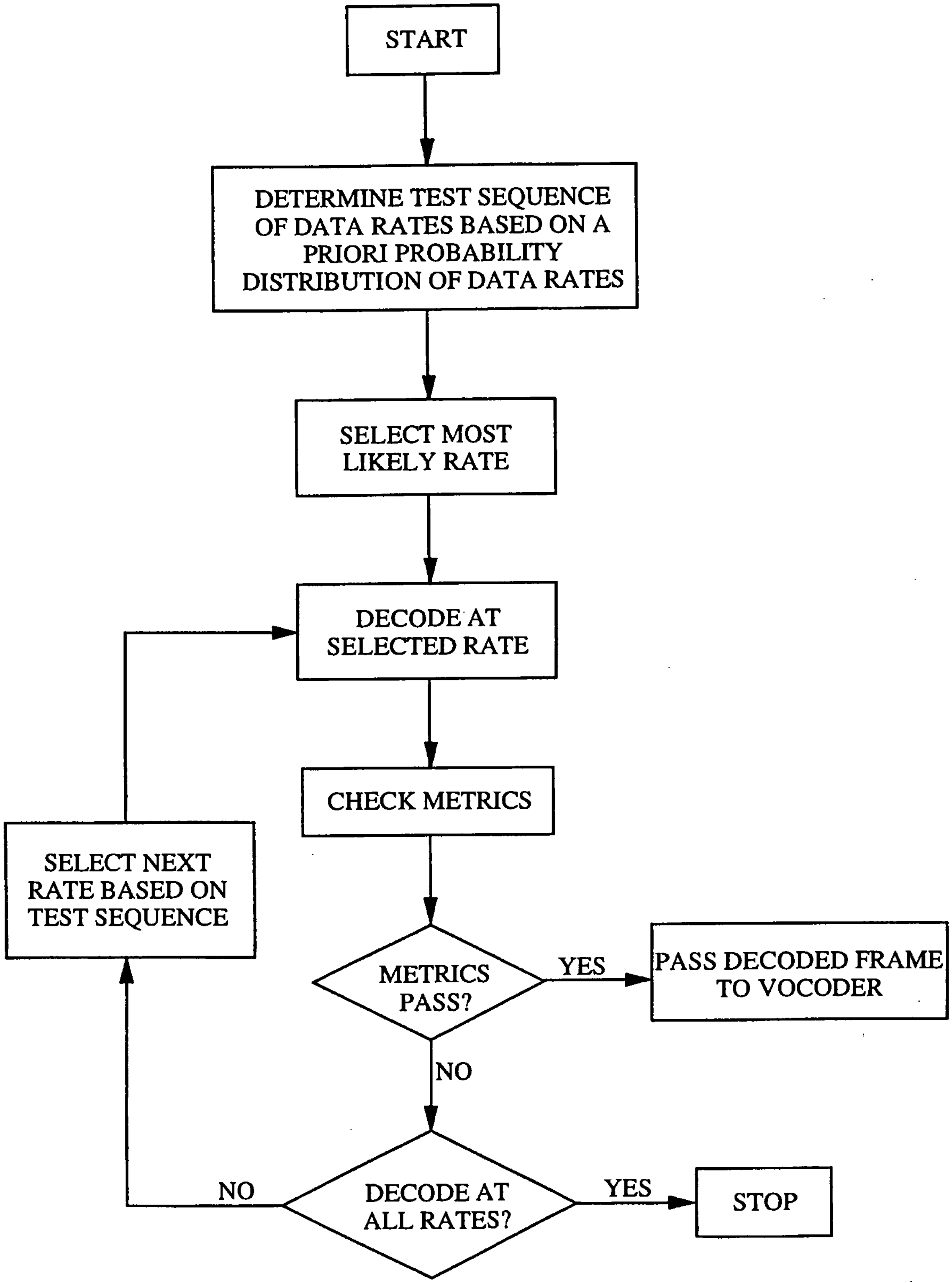

START
DETERMINE TEST SEQUENCE OF DATA RATES BASED ON A PRIORI PROBABILITY DISTRIBUTION OF DATA RATES
SELECT MOST LIKELY RATE
DECODE AT SELECTED RATE
CHECK METRICS
METRICS PASS?
YES
PASS DECODED FRAME TO VOCODER
NO
DECODE AT ALL RATES?
YES
STOP
NO
SELECT NEXT RATE BASED ON TEST SEQUENCE

FIG. 5

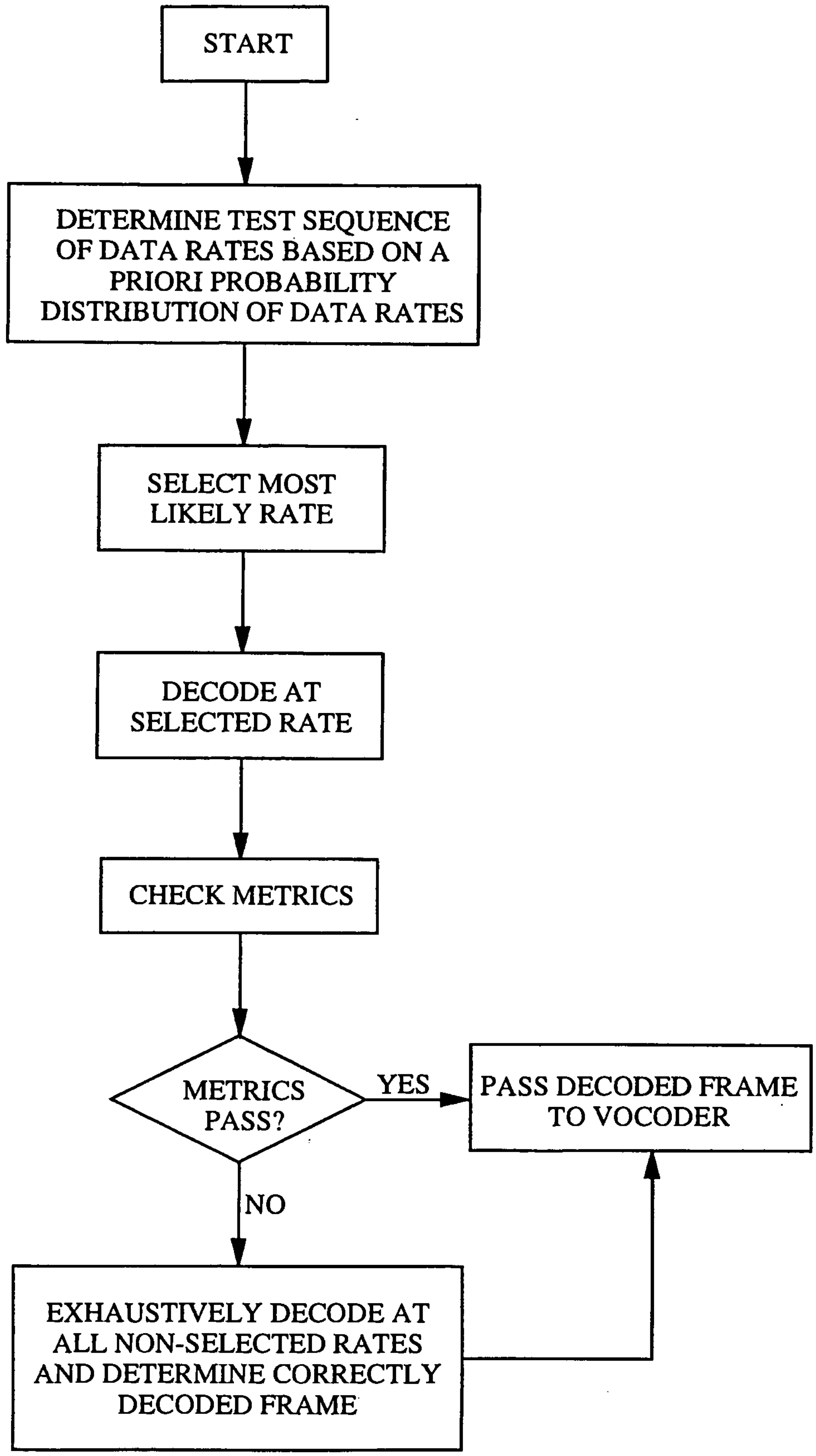

START
DETERMINE TEST SEQUENCE OF DATA RATES BASED ON A PRIORI PROBABILITY DISTRIBUTION OF DATA RATES
SELECT MOST LIKELY RATE
DECODE AT SELECTED RATE
CHECK METRICS
METRICS PASS?
YES
PASS DECODED FRAME TO VOCODER
NO
EXHAUSTIVELY DECODE AT ALL NON-SELECTED RATES AND DETERMINE CORRECTLY DECODED FRAME

FIG. 6

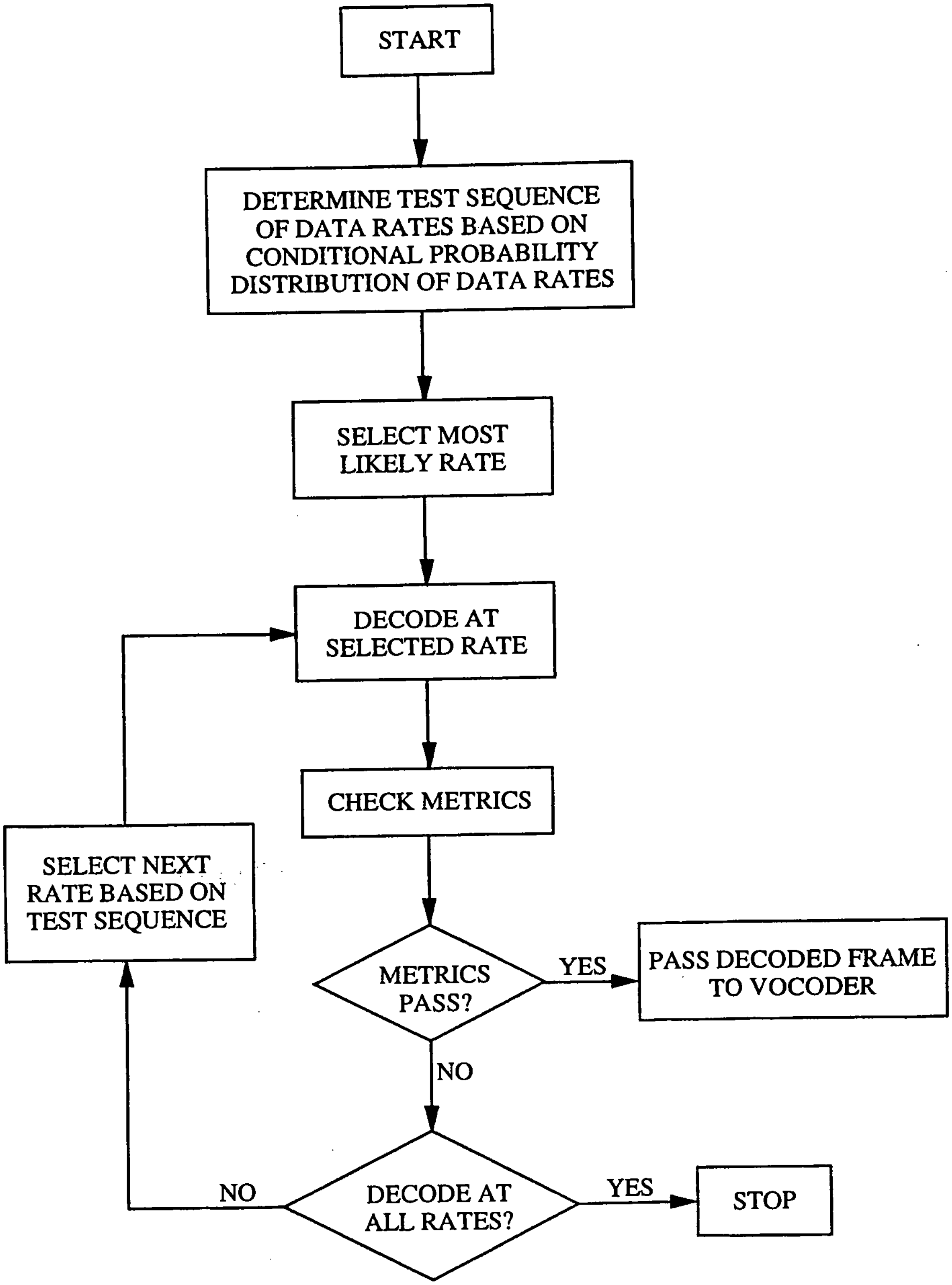

START
DETERMINE TEST SEQUENCE OF DATA RATES BASED ON CONDITIONAL PROBABILITY DISTRIBUTION OF DATA RATES
SELECT MOST LIKELY RATE
DECODE AT SELECTED RATE
CHECK METRICS
METRICS PASS?
YES
PASS DECODED FRAME TO VOCODER
NO
DECODE AT ALL RATES?
YES
STOP
NO
SELECT NEXT RATE BASED ON TEST SEQUENCE

FIG. 7

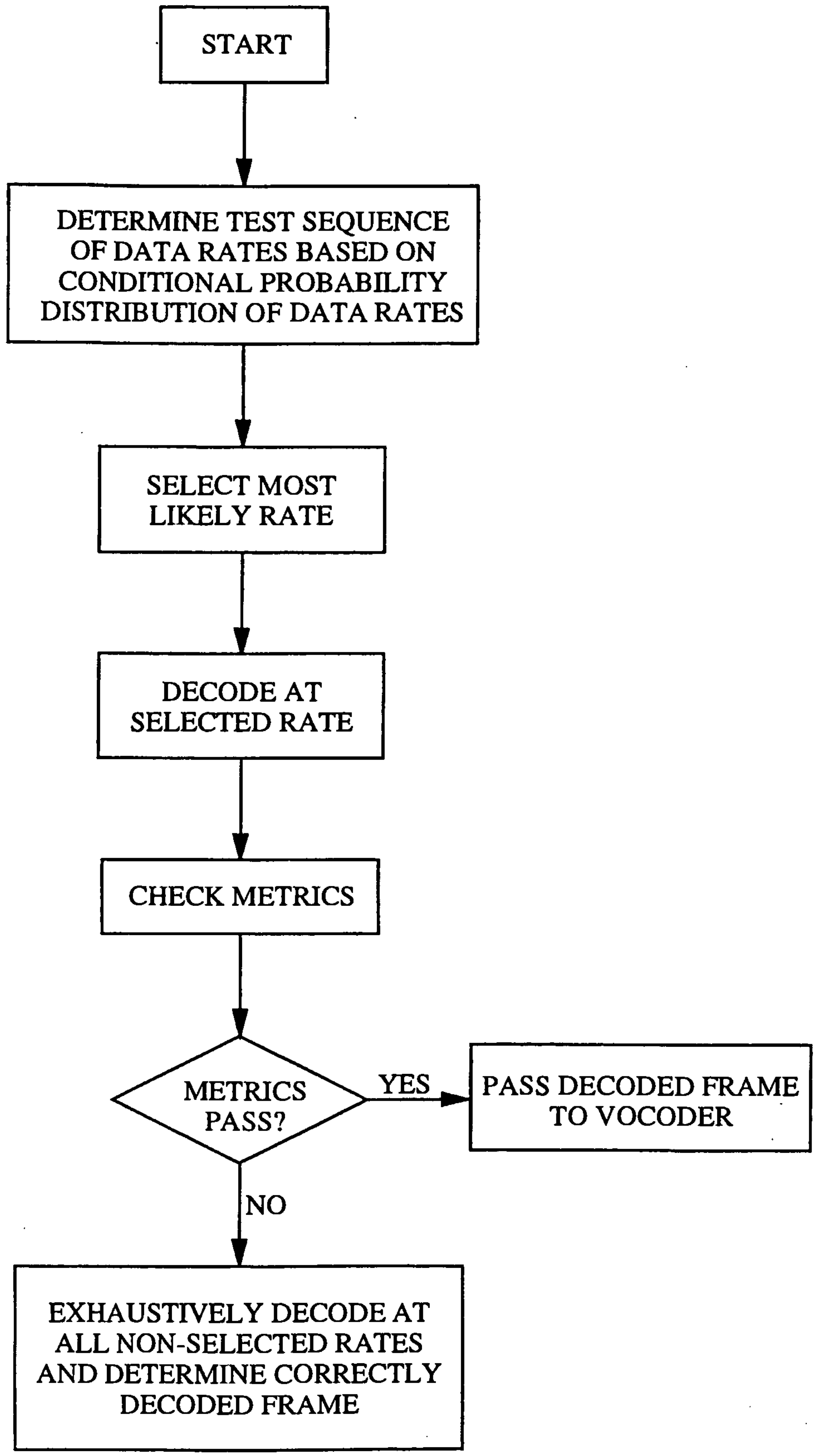

START
DETERMINE TEST SEQUENCE OF DATA RATES BASED ON CONDITIONAL PROBABILITY DISTRIBUTION OF DATA RATES
SELECT MOST LIKELY RATE
DECODE AT SELECTED RATE
CHECK METRICS
METRICS PASS?
YES
PASS DECODED FRAME TO VOCODER
NO
EXHAUSTIVELY DECODE AT ALL NON-SELECTED RATES AND DETERMINE CORRECTLY DECODED FRAME

FIG. 8

INTERNATIONAL SEARCH REPORT

Inte. ıonal Application No
PCT/US 97/19676

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L25/02

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 6 H04L

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.
A	EP 0 711 056 A (NIPPON ELECTRIC CO) 8 May 1996 see abstract see page 2, column 2, line 46 - page 3, column 3, line 26 see page 4, column 5, line 13 - column 6, line 23; figure 2 ---	1,14,15, 22,27,28
A	EP 0 713 305 A (NIPPON ELECTRIC CO) 22 May 1996 see abstract see page 2, column 2, line 45 - page 3, column 3, line 12 see page 3, column 3, line 33 - column 4, line 2; figure 1 see page 3, column 4, line 44 - page 4, column 5, line 13 --- -/--	1,14,15, 22,27,28

[X] Further documents are listed in the continuation of box C.

[X] 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

1

Date of the actual completion of the international search	Date of mailing of the international search report
27 February 1998	09/03/1998
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 Bossen, M

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte .onal Application No
PCT/US 97/19676

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 566 206 A (BUTLER BRIAN K ET AL) 15 October 1996 cited in the application see column 2, line 37 - line 47 see column 5, line 57 - column 6, line 32; figure 2 see column 7, line 35 - line 52; figure 4 -----	7,9-12, 14, 22-25,27

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte .ional Application No
PCT/US 97/19676

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0711056 A	08-05-96	JP 8130535 A	21-05-96
		AU 3662595 A	09-05-96
		CA 2163134 A	03-05-96
EP 0713305 A	22-05-96	JP 2596392 B	02-04-97
		JP 8149567 A	07-06-96
		AU 686026 B	29-01-98
		AU 3787595 A	23-05-96
		CA 2162417 A	17-05-96
		FI 955398 A	17-05-96
US 5566206 A	15-10-96	AT 158910 T	15-10-97
		AU 683479 B	13-11-97
		AU 7113694 A	17-01-95
		BR 9406891 A	26-03-96
		DE 69405997 D	06-11-97
		EP 0705512 A	10-04-96
		FI 956091 A	16-02-96
		JP 9501548 T	10-02-97
		WO 9501032 A	05-01-95
		CN 1108834 A	20-09-95
		IL 109842 A	30-09-97
		MX 9404610 A	31-01-95
		ZA 9404032 A	09-03-95

CORRECTED VERSION*

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/00

A1

(11) International Publication Number: WO 98/11690

(43) International Publication Date: 19 March 1998 (19.03.98)

(21) International Application Number: PCT/US97/16223

(22) International Filing Date: 12 September 1997 (12.09.97)

(30) Priority Data:

60/025,991	12 September 1996 (12.09.96)	US
08/887,723	3 July 1997 (03.07.97)	US

(71)(72) Applicant and Inventor: GLOVER, John, J. [US/US]; 26 Amaranth Avenue, Medford, MA 02155 (US).

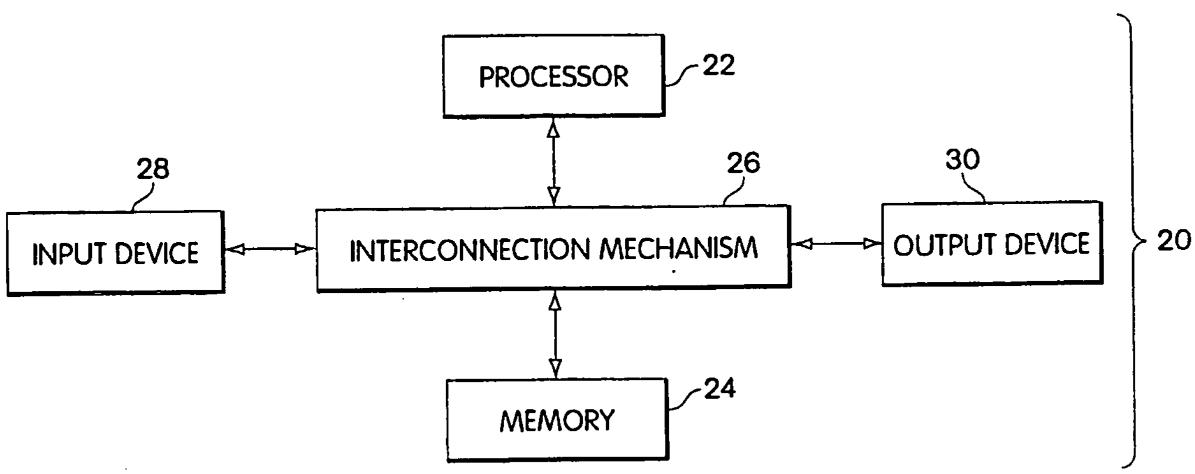
(74) Agent: GORDON, Peter, J.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, 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, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, 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

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.

(54) Title: SELF-DECRYPTING DIGITAL INFORMATION SYSTEM AND METHOD



(57) Abstract

The claimed data protection device (20) includes a processor (22) connected to a memory system (24) through an interconnection mechanism (26). An input device (28) is also connected to the processor (22) and memory system (24) through the interconnection mechanism (26). The interconnection mechanism (26) is typically a combination of one or more buses and one or more switches. The output device (30) may be a display, and the input device (28) may be a keyboard and/or mouse or other cursor control device.

*(Referred to in PCT Gazette No. 32/1998, 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.

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		

SELF-DECRYPTING DIGITAL INFORMATION SYSTEM AND METHOD

Field of the Invention

The present invention is related to mechanisms for protecting digital information from being copied. In particular, the present invention is related to mechanisms which permit authorized execution of computer program code or access to other digital information which is encrypted or otherwise encoded.

Background of the Invention

A serious problem which faces the electronic publishing and software industries is the ease with which digital information can be copied without authorization from the publisher. Digital information also may be used or modified without authorization. For example, computer software may be reverse engineered or attacked by computer viruses.

There are many mechanisms available which may be used to limit or prevent access to digital information. Such mechanisms often either restrict the ability of the user to make back-up copies or involve the use of special purpose hardware to limit access to the digital information. For example, some mechanisms restrict the use of digital information to a particular machine. See, for example, U.S. Patent 4,817,140. Other mechanisms require the digital information to be stored on a particular recording medium in order to be used. See, for example, U.S. Patent 5,412,718. Yet other mechanisms allow only a certain number of uses of the digital information. See for example, U.S. Patent 4,888,798. Many of these access control mechanisms cause distribution to be more costly.

Several other patents describe a variety of systems for encryption, compression, licensing and royalty control and software distribution such as: U.S. Pat. No. 4,405,829, U.S. Pat. No. 4,864,616, U.S. Pat. No. 4,888,800, U.S. Pat. No. 4,999,806, U.S. Pat. No. 5,021,997, U.S. Patent No. 5,027,396, U.S. Pat. No. 5,033,084, U.S. Pat. No. 5,081,675, U.S. Pat. No. 5,155,847, U.S. Pat. No. 5,166,886, U.S. Pat. No. 5,191,611, U.S. Pat. No. 5,220,606, U.S. Pat. No. 5,222,133, U.S. Pat. No. 5,272,755, U.S. Pat. No. 5,287,407, U.S. Pat. No. 5,313,521, U.S. Pat. No. 5,325,433, U.S. Pat. No. 5,327,563, U.S. Pat. No. 5,337,357, U.S. Pat. No. 5,351,293, U.S. Pat. No. 5,341,429, U.S. Pat. No. 5,351,297, U.S. Pat. No. 5,361,359, U.S. Pat. No. 5,379,433, U.S. Pat. No. 5,392,351, U.S. Pat. No. 5,394,469, U.S. Pat. No. 5,414,850, U.S. Pat. No. 5,473,687, U.S. Pat. No. 5,490,216, U.S. Pat. No. 5,497,423, U.S. Pat. No. 5,509,074, U.S.

Pat. No. 5,,511,123, U.S. Pat. No. 5,524,072, U.S. Pat. No. 5,532,920, U.S. Pat. No. 5,555,304, U.S. Pat. No. 5,557,346, U.S. Pat. No. 5,557,765, U.S. Pat. No. 5,592,549, U.S. Pat. No. 5,615,264, U.S. Pat. No. 5,625,692, and U.S. Pat. No. 5,638,445.

Computer programs or other digital information also may be encrypted in order to prevent an individual from making a useful copy of the information or from reverse engineering a program. Even with such encryption, however, a computer program must be decrypted in order for a computer to load and execute the program. Similarly, other digital information must be decrypted before it can be accessed and used. Generally, digital information is decrypted to disk, and not to main memory of the computer which is more protected by the operating system, because decryption to main memory results in a significant loss of memory resources. If the purpose for using encryption is to prevent users from copying the digital information, then decryption of the information to accessible memory for use defeats this purpose.

One way to protect digital information using encryption has been made available by International Business Machines (IBM) and is called a "CRYPTOLOPE" information container. This technology is believed to be related to U.S. Patent Nos. 5,563,946 and 5,598,470 (to Cooper et al.), and published European patent applications 0679977, 0679978, 0679979 and 0681233. The CRYPTOLOPE system requires a user to have a "helper application" and a key. The CRYPTOLOPE information container is generated by IBM. The content provider submits data to IBM, which in turn encrypts and packages the data in a CRYPTOLOPE information container. The helper application is a form of memory resident program, called a terminate and stay resident (TSR) program, which is a form of input/output (I/O) device driver installed in the operating system and which monitors requests from the operating system for files on specified drives and directories. Because the TSR program must know the directory, and/or file name to be accessed, that information also is available to other programs. Other programs could use that information to manipulate the operation of the TSR program in order to have access to decrypted contents of the information container. The encrypted information container includes an executable stub which is executed whenever the application is run without the installed TSR program or from a drive not monitored by the TSR program to prevent unpredictable activity from executing encrypted code. This stub may be used to install decryption and cause the application be executed a second time, or to communicate with the TSR program to instruct the TSR program to monitor the drive. It may be preferable from the point of view of the content provider however to maintain an encryption process and keys independently of any third party.

Multimedia content, such as a movie or hypertext presentation also may be stored on a digital versatile disk (DVD), sometimes called a digital video disk, compact disk read-only memory (CD-ROM), rewriteable compact disks (CD-RW) or other medium in an encrypted digital format for use with special-purpose devices. For example, concern about illegal copying of content from digital video disks or other digital media has resulted in a limited amount of content being available for such devices. This problem has caused representatives of both multimedia providers and digital video disk manufacturers to negotiate an agreement on an encryption format for information stored on DVDs. This copy protection scheme is licensed through an organization called the CSS Interim Licensing organization. However, in this arrangement, the content provider is limited to using the agreed upon encryption format and a device manufacturer is limited to using a predetermined decryption system.

Encryption has also been used to protect and hide computer viruses. Such viruses are typically polymorphic, i.e., they change every time they infect a new program, and are encrypted. The virus includes a decryption program that executes to decrypt the virus every time the infected program is run. Such viruses are described, for example, in "Computer Virus-Antivirus Coevolution" by Carey Nachenberg, Communications of the ACM, Vol. 40, No. 1, (Jan. 1997), p. 46 et seq. Such viruses include decryption keys within them since, clearly, their execution is not carried out by the user and a user would not be asked for authorization keys to permit execution of the viruses. Additionally, such viruses are typically only executed once at the start of execution of an infected program and permanently return control to the infected program after execution.

Summary of the Invention

Some of these problems with digital information protection systems may be overcome by providing a mechanism which allows a content provider to encrypt digital information without requiring either a hardware or platform manufacturer or a content consumer to provide support for the specific form of corresponding decryption. This mechanism can be provided in a manner which allows the digital information to be copied easily for back-up purposes and to be transferred easily for distribution, but which should not permit copying of the digital information in decrypted form. In particular, the encrypted digital information is stored as an executable computer program which includes a decryption program that decrypts the encrypted information to provide the desired digital information, upon successful completion of an authorization procedure by the user.

In one embodiment, the decryption program is executed as a process within a given operating system and decrypts the digital information within the memory area assigned to that process. This memory area is protected by the operating system from copying or access by other processes. Even if access to the memory area could be obtained, for example through the operating system, when the digital information is a very large application program or a large data file, a copy of the entire decrypted digital information is not likely to exist in the memory area in complete form.

By encrypting information in this manner, a platform provider merely provides a computer system with an operating system that has adequate security to define a protected memory area for a process and adequate functionality to execute a decryption program. The content provider in turn may use any desired encryption program. In addition, by having a process decrypt information within a protected memory area provided by the operating system, the decrypted information does not pass through any device driver, memory resident program or other known logical entity in the computer system whose behavior may be controlled to provide unauthorized access to the data. The ability to reverse engineer or attack a computer program with a computer virus also may be reduced.

In another embodiment, the decryption program is part of a dynamically loaded device driver that responds to requests for data from the file containing the encrypted data. When the digital information product is first executed, this device driver is extracted from the file and is loaded into the operating system. The executed digital information product then informs the loaded device driver of the location of the hidden information in the file, any keys or other passwords, and the name of a phantom directory and file to be called that only the digital information product and the device driver know about. The name of this directory may be generated randomly. Each segment of hidden information in the digital information product may be assigned its own unique file name in the phantom directory. The digital information product then makes a call to the operating system to execute one of the files in the phantom directory. The loaded driver traps these calls to the operating system, accesses the original file, decrypts the desired information and outputs the desired information to the operating system.

In combination with other mechanisms that track distribution, enforce royalty payments and control access to decryption keys, the present invention provides an improved method for identifying and detecting sources of unauthorized copies. Suitable authorization procedures also enable the digital information to be distributed for a limited number of uses and/or users, thus enabling per-use fees to be charged for the digital information.

Accordingly, one aspect of the invention is a digital information product including a computer-readable medium with digital information stored thereon. The digital information includes computer program logic having a first portion of executable computer program logic and a second portion of digital information. The first portion of executable program logic, when executed, defines a mechanism for responding to requests for digital information from an operating system of a computer. This mechanism, when used to access the second portion of the encrypted digital information, decrypts the encrypted digital information, and provides the encrypted digital information to the operating system.

In the foregoing aspect of the invention, the digital information may be executable computer program logic. Hence, one aspect of the invention is a computer program product, including a computer readable medium with computer program logic stored thereon. The computer program logic includes a first portion of executable computer program logic and a second portion of encrypted computer program logic. The first portion of executable computer program logic, when executed, defines a mechanism for responding to requests for computer program logic from an operating system of a computer. This mechanism accesses the second portion of encrypted computer program logic, decrypts the encrypted computer program logic, and provides the decrypted computer program logic to the operating system.

Another aspect of the present invention is a computer program product, a computer system and a process which produce a computer program or digital information product in accordance with other aspects of the invention, using executable program code for the first and second portions of the desired computer program product.

Another aspect of the present invention is a computer program product including a self-decrypting encrypted executable computer program. The product includes a computer readable medium having computer program logic stored thereon. The computer program logic defines first, second and third modules, wherein the third module defines the encrypted executable computer program. The first module, when executed by a computer, defines a mechanism for loading the second module into memory of the computer. The second module, when executed by a computer, defines a mechanism for communicating with an operating system of the computer to receive requests for program code from the encrypted executable computer program from the third module, and for processing the requests to access and decrypt the encrypted executable computer program and for providing the decrypted executable code from the third module to the operating system.

Another aspect of the invention is a process for executing encrypted executable computer programs on a computer system having a processor, memory and operating system. The process involves receiving computer program logic having a first module defining a start up routine, a second module, and a third module containing the encrypted executable computer program. The first module of the received computer program logic is executed using the processor. When the first module is executed, the second module is caused to be loaded into the memory of the computer system. Requests are generated from the operating system for data from the encrypted executable computer program and are received by the second module. The second module accesses and decrypts the encrypted executable computer program in response to these requests and returns the decrypted executable computer program to the operating system.

These and other aspects, advantages and features of the present invention and its embodiments will be more apparent given the following detailed description.

Brief Description of the Drawing

In the drawing,

Fig. 1 is a block diagram of a typical computer system with which the present invention may be implemented;

Fig. 2 is a block diagram of a memory system in the computer system of Fig. 1;

Fig. 3 is a diagram of a computer program or digital information product which may be recorded on a computer readable and writable medium, such as a magnetic disc;

Fig. 4 is a flowchart describing how the computer program or digital information product of Fig. 3 is used;

Fig. 5 is a flowchart describing operation of an example unwrap procedure as shown in Fig. 3 in one embodiment of the invention;

Fig. 6 is a flowchart describing operation of an example device driver as shown in Fig. 3 in one embodiment of the invention;

Fig. 7 is a block diagram of a computer system in the process of executing a computer program product in accordance with one embodiment of the invention;

Fig. 8 is a flowchart describing operation of an example unwrap procedure in another embodiment of the invention; and

Fig. 9 is a flowchart describing how a computer program product such as shown in Fig. 3 is constructed.

Detailed Description

The present invention will be more completely understood through the following detailed description which should be read in conjunction with the attached drawing in which similar reference numbers indicate similar structures.

Embodiments of the present invention may be implemented using a general purpose digital computer or may be implemented for use with a digital computer or digital processing circuit. A typical computer system 20 is shown in Fig. 1, and includes a processor 22 connected to a memory system 24 via an interconnection mechanism 26. An input device 28 also is connected to the processor and memory system via the interconnection mechanism, as is an output device 30. The interconnection mechanism 26 is typically a combination of one or more buses and one or more switches. The output device 30 may be a display and the input device may be a keyboard and/or a mouse or other cursor control device.

It should be understood that one or more output devices 30 may be connected to the computer system. Example output devices include a cathode ray tube (CRT) display, liquid crystal display (LCD), television signal encoder for connection to a television or video tape recorder, printers, communication devices, such as a modem, and audio output. It also should be understood that one or more input devices 28 may be connected to the computer system. Example input devices include a keyboard, keypad, trackball, mouse, pen and tablet, communication device, audio or video input and scanner. It should be understood that the invention is not limited to the particular input or output devices used in combination with the computer system or to those described herein.

The computer system 20 may be a general purpose computer system, which is programmable using a high level computer programming language, such as "C++," "Pascal," "VisualBasic." The computer system also may be implemented using specially programmed, special purpose hardware. In a general purpose computer system, the processor is typically a commercially available processor, such as the Pentium processor from Intel Corporation. Many other processors are also available. Such a processor executes a program called an operating system, such as Windows 95 or Windows NT 4.0, both available from Microsoft Corporation, which controls the execution of other computer programs and provides scheduling, debugging, input output control, accounting compilation, storage assignment, data management and memory management, and communication control and related services. Other examples of operating systems include: MacOS System 7 from Apple Computer, OS/2 from IBM, VMS from Digital Equipment Corporation, MS-DOS from Microsoft Corporation, UNIX from AT&T, and IRIX from Silicon Graphics, Inc.

The computer system 20 also may be a special purpose computer system such as a digital versatile disk or digital video disk (DVD) player. In a DVD player, there is typically a decoder controlled by some general processor which decodes an incoming stream of data from a DVD. In some instances, the DVD player includes a highly integrated DVD decoder engine. Such devices generally have a simple operating system which may be modified to include the capabilities described and used herein in connection with the typical operating systems in a general purpose computer. In particular, some operating systems are designed to be small enough for installation in an embedded system such as a DVD player, including the WindowsCE operating system from Microsoft Corporation and the JavaOS operating system from SunSoft Corporation. The operating system allows a content provider to provide its own programs that define some of the content, which is particularly useful for interactive multimedia. This capability also can be used to provide encryption and decryption, in accordance with the invention.

The processor and operating system define a computer platform for which application programs in a programming language such as an assembly language or a high level programming language are written. It should be understood that the invention is not limited to a particular computer platform, operating system, processor, or programming language. Additionally, the computer system 20 may be a multi-processor computer system or may include multiple computers connected over a computer network.

An example memory system 24 will now be described in more detail in connection with Fig. 2. A memory system typically includes a computer readable and writable non-volatile recording medium 40, of which a magnetic disk, a flash memory, rewriteable compact disk (CD-RW) and tape are examples. The recording medium 40 also may be a read only medium such as a compact disc-read only memory (CD-ROM) or DVD. A magnetic disk may be removable, such as a "floppy disk" or "optical disk," and/or permanent, such as a "hard drive." The disk, which is shown in Fig. 2, has a number of tracks, as indicated at 42, in which signals are stored, in binary form, i.e., a form interpreted as a sequence of 1's and 0's, as shown at 44. Such signals may define an application program to be executed by the microprocessor, or information stored on the disk to be processed by the application program. Typically, in the operation of a general purpose computer, the processor 22 causes data to be read from the non-volatile recording medium 40 into an integrated circuit memory element 46, which is typically a volatile random access memory, such as a dynamic random access memory (DRAM) or static random access memory (SRAM). The integrated circuit memory element 46 allows for faster access to the information by the processor than disk 40, and is typically called the system or host memory. The processor generally causes the data to be manipulated within the integrated circuit memory 46 and may copy the data to the disk 40, if modified, when processing is completed. A variety of mechanisms are known for managing data movement between the disk 40 and the integrated circuit memory 46, and the invention is not limited thereto. It should also be understood that the invention is not limited to a particular memory system.

The file system of a computer generally is the mechanism by which an operating system manages manipulation of data between primary and secondary storage, using files. A file is a named logical construct which is defined and implemented by the operating system to map the name and a sequence of logical records of data to physical storage media. An operating system may specifically support various record types or may leave them undefined to be interpreted or controlled by application programs. A file is referred to by its name by application programs and is accessed through the operating system using commands defined by the operating system. An operating system provides basic file operations provided by for creating a file, opening a file, writing a file, reading a file and closing a file.

In order to create a file, the operating system first identifies space in the storage media which is controlled by the file system. An entry for the new file is then made in a directory which includes entries indicating the names of the available files and their locations in the file system. Creation of a file may include allocating certain available space to the file. Opening a file returns a handle to the application program which it uses to access the file. Closing a file invalidates the handle.

In order to write data to a file, an application program issues a command to the operating system which specifies both an indicator of the file, such as a file name, handle or other descriptor, and the information to be written to the file. Given the indicator of the file, the operating system searches the directory to find the location of the file. The directory entry stores a pointer, called the write pointer, to the current end of the file. Using this pointer, the physical location of the next available block of storage is computed and the information is written to that block. The write pointer is updated in the directory to indicate the new end of the file.

In order to read data from a file, an application program issues a command to the operating system specifying the indicator of the file and the memory locations assigned to the application where the next block of data should be placed. The operating system searches its directory for the associated entry given the indicator of the file. The directory may provide a pointer to a next block of data to be read, or the application may program or specify some offset from the beginning of the file to be used.

A primary advantage of using a file system is that, for an application program, the file is a logical construct which can be created, opened, written to, read from and closed without any concern for the physical storage used by the operating system.

The operating system also allows for the definition of another logical construct called a process. A process is a program in execution. Each process, depending on the operating system, generally has a process identifier and is represented in an operating system by a data structure which includes information associated with the process, such as the state of the process, a program counter indicating the address of the next instruction to be executed for the process, other registers used by process and memory management information including base and bounds registers. Other information also may be provided. The base and bounds registers specified for a process contain values representing the largest and smallest addresses that can be generated and accessed by an individual program. Where an operating system is the sole entity able to modify these memory management registers, adequate protection from access to the memory locations of one process from another process is provided. As a result, this memory management information is used by the operating system to provide a protected memory area for the process. A process generally uses the file system of the operating system to access files.

The present invention involves storing encrypted digital information, such an audio, video, text or an executable computer program, on a computer readable medium such that it can be copied easily for back-up purposes and transferred easily for distribution, but also such that it cannot be copied readily in decrypted form during use. In particular, the digital information is stored as a computer program that decrypts itself while it is used to provide the digital information, e.g., to provide executable operation code to the operating system of a computer, as the digital information is needed. Any kind of encryption or decryption may be used and also may include authorization mechanisms and data compression and decompression. In one embodiment of the present invention, decrypted digital information exists only in memory accessible to the operating system and processes authorized by the operating system. When the digital information is a large application program, a copy of the entire decrypted application program is not likely to exist in the main memory at any given time, further reducing the likelihood that a useful copy of decrypted code could be made. The decryption operation also is performed only if some predetermined authorization procedure is completed successfully.

One embodiment of the invention, in which the decryption program is a form of dynamically loaded device driver, will first be described. Fig. 3 illustrates the structure of digital information as stored in accordance with one embodiment of the present invention, which may be stored on a computer readable medium such as a magnetic disc or compact disc read only memory (CD-ROM) to form a computer program product. The digital information includes a first portion 50, herein called an unwrap procedure or application, which is generally unencrypted executable program code. The purpose of the unwrap procedure is to identify the locations of the other portions of the digital information, and may perform other operations such as verification. In particular, the unwrap procedure identifies and extracts a program which will communicate with the operating system, herein called a virtual device driver 52. The unwrap procedure may include decryption and decompression procedures to enable it to decrypt/decompress the driver, and/or other content of this file. The program 52 need not be a device driver. The virtual device driver 52 typically follows the unwrap procedure 50 in the file container, the digital information. The virtual device driver, when executed, decrypts and decodes the desired digital information such as an executable computer program code from hidden information 54, which may be either encrypted and/or encoded (compressed). It is the decrypted hidden information which is the desired digital information to be accessed. This hidden information may be any kind of digital data, such as audio, video, text, and computer program code including linked libraries or other device drivers.

In this embodiment of the computer program product, labels delineate the boundaries between the device driver and the hidden files. These labels may or may not be encrypted. A first label 56 indicates the beginning of the code for the virtual device driver 52. A second label 58 indicates the end of the virtual device driver code. Another label 60 indicates the beginning of the hidden information and a label 62 indicates the end of that application. There may be one or more blocks of such hidden information, each of which can be given a different name. It may be advantageous to use the name of the block of information in its begin and end tags. This computer program product thus contains and is both executable computer program code and one or more blocks of digital information. A table of locations specifying the location of each portion of the product could be used instead of labels. Such a table could be stored in a predetermined location and also may be encrypted.

The overall process performed using this computer program product in one embodiment of the invention will now be described in connection with Fig. 4. This embodiment may be implemented for use with the Windows95 operating system and is described in more detail in connection with Figs. 5-7. An embodiment which may be implemented for use on the WindowsNT 4.0 operating system is described in more detail below in connection with Fig. 8. In both of these described embodiments, the digital information is an executable computer program which is read by the operating system as data from this file and is executed. The same principle of operation would apply if the data were merely audio, video, text or other information to be conveyed by a user. In the embodiment of Fig. 4, the computer program is first loaded into memory in step 70, and the unwrap procedure 50 is executed by the operating system, as any typical executable computer program is executed. The unwrap procedure may perform authorization, for example by checking for a required password or authentication code, and may receive any data needed for decryption or decompression, for example keys or passwords, in step 72. Suitable authorization procedures may provide the ability to distribute software for single use. The unwrap procedure locates the virtual device driver 52 within the computer program in step 74, and then locates the hidden application in step 76. The virtual device driver 52 is then extracted by the unwrap procedure from the computer program, copied to another memory location and loaded for use by the operating system in step 78. An advantage of an operating system like Windows95 is that it allows such device drivers to be loaded dynamically without restarting the computer.

The executed unwrap procedure 50, in step 80, informs the loaded virtual device driver 52 of the location of the hidden information in the file, any keys or other passwords, and a name of a phantom directory and file to be called that only the unwrap procedure and the virtual device driver know about. The name of this phantom directory may be generated randomly. Each segment information hidden in the digital information product may be assigned its own unique file name in the phantom directory.

After the loaded virtual device driver 52 receives all communications from the unwrap procedure, it opens the original application file for read only access in step 82. The unwrap procedure then makes a call to the operating system in step 84 to execute the file in the phantom directory for which the name was transmitted to the loaded virtual device driver. One function of the loaded virtual device driver 52 is to trap all calls from the operating system to access files in step 86. Any calls made by the operating system to access files in the phantom directory are processed by the virtual device driver, whereas calls to access files in other directories are allowed to proceed to their original destination. In response to each call from the operating system, the virtual device driver obtains the bytes of data requested by the operating system from the original computer program file in step 88. These bytes of data are then decrypted or decompressed in step 90 and returned to the operating system. When processing is complete, the phantom application is unloaded from the operating system in step 92, and may be deleted from the memory.

A more detailed description of the process of Fig. 4 will now be described in connection with Figs. 5-7. Fig. 5 is a flowchart describing the operation of one embodiment of the unwrap procedure in more detail. The first step performed by this procedure is identifying the operating system being used, in step 100. This step is useful because different methods may be used with different operating systems. All code that may be used to run in various operating systems may be placed in this unwrap procedure. This procedure also may contain the decompression/decryption code, for example or any other computer program code to be executed.

The executed application then opens the original executable file as a data file and searches for the begin and end tags of the device driver and hidden files in step 102. The device driver code is copied into memory and loaded into the operating system in step 104. The unwrap procedure then informs the device driver of the name of the original application file, offsets of the hidden files and the name of a phantom directory, which is typically randomly generated (step 106). This communication may be performed using a "DeviceIOControl" function call in the Windows95 operating system. The unwrap procedure then makes a call to the operating system to execute the hidden file in the phantom directory, in step 108.

The operation of one embodiment of a device driver will now be described in connection with Fig. 6. After the device driver is loaded into the operating system, it hooks into a position between the operating system and a file system driver (FSD), in step 110, to intercept calls made by the operating system to the FSD for data from files in the phantom directory. The FSD is the code within the operating system that performs physical reading and writing of data to disk drives. The operating system makes requests to the FSD for data from files in directories on the disk drives. The driver then receives information from the unwrap procedure including the name of the original file, the location of hidden files within the original file, and the name of the phantom directory created by the unwrap procedure (step 112). The device driver opens the original file as a read only data file. The device driver now traps calls, in step 114, made from the operating system for files in the phantom directory. Calls to other directories are ignored and passed on to the original destination. The device driver then reads the data from the original data file, decrypts and decompresses it, and returns the decrypted/decompressed data to the operating system in step 116.

For example, if the offset for the hidden application in the original data file is 266,270 bytes and the operating system asks for 64 bytes starting at offset 0 of the hidden application in the phantom directory, the device driver reads 64 bytes from the original file starting at offset 266,270, decrypts/decompresses those 64 bytes, and returns the first 64 decrypted/decompressed bytes back to the operating system. From the point of view of the operating system, the 64 bytes appear to have come from the file in the phantom directory. Steps 114 and 116 are performed on demand in response to the operating system.

A block diagram of the computer system in this embodiment, with a device driver loaded and in operation, will now be described in more detail in connection with Fig. 7. Fig. 7 illustrates the operating system 120, the loaded device driver 122, a file system driver 124, the original executable file 126 as it may appear on disk and the unwrap procedure 128. The executable file may in fact be on a remote computer and accessed through a network by the device driver. The unwrap procedure causes the operating system to begin execution of the hidden file by issuing an instruction to execute the file in the phantom directory, as indicated at 130. This command is issued after the device driver 122 is informed of the file name of the original executable file 126, offsets of the hidden files within that file and the name of the phantom directory, as indicated at 132. The operating system then starts making calls to the phantom directory as indicated at 134. The device driver 122 traps these calls and turns them into requests 136 to the file system driver to access the original executable file 126. Such requests actually are made to the operating system 120, through the device driver 122 to the file system driver 124. The file system driver 124 returns encrypted code 138 to the device driver

122. The encrypted code 138 actually passes back through the device driver 122 to the operating system 120 which in turn provides the encrypted code 138 to the device driver 122 as the reply to the request 136 for the original file. The device driver 122 then decrypts the code to provide decrypted code 140 to the operating system 120.

Another embodiment of the invention will now be described in connection with Fig. 8. This embodiment may be implemented using the WindowsNT 4.0 operating system, for example. In this embodiment, the device driver portion 52 of the computer program product is not used. The unwrap procedure for this embodiment begins by identifying the operating system being used similar, which is step 100 in Fig. 5. If the operating system is Windows NT 4.0, for example, a different unwrap procedure for this embodiment is performed. Before describing this unwrap procedure, a brief description of some of the available operating system commands will be provided.

Currently, under all versions of the Window operating system or operating environment from Microsoft Corporation (such as Windows 3.1, Windows 95 and Windows NT 3.51 and 4.0) all executable files (.exe) or dynamic link library (.dll and .ocx) files, which are executable files with different header and loading requirements than .exe files, that are loaded into memory by the operating system must reside as a file either locally, e.g., on a disk drive or remotely, e.g., over a network or communications port. All further references herein to loading an executable will be using the Win32 function calls used in Windows 95 and NT 3.51 and 4.0 operating systems. The CreateProcess() function which loads files with an .exe extension takes ten parameters:

```
BOOL CreateProcess(// Prototype from Microsoft Visual C++ Help Documentation
  LPCTSTR lpApplicationName,                          // pointer to name of executable module
  LPTSTR lpCommandLine,                               // pointer to command line string
  LPSECURITY_ATTRIBUTES lpProcessAttributes,          // pointer to process security attributes
  LPSECURITY_ATTRIBUTES lpThreadAttributes,           // pointer to thread security attributes
  BOOL bInheritHandles,                               // handle inheritance flag
  DWORD dwCreationFlags,                              // creation flags
  LPVOID lpEnvironment,                               // pointer to new environment block
  LPCTSTR lpCurrentDirectory,                         // pointer to current directory name
  LPSTARTUPINFO lpStartupInfo,                        // pointer to STARTUPINFO
  LPPROCESS_INFORMATION lpProcessInformation          // pointer to PROCESS_INFORMATION
);
```

Three of these parameters are pointers to strings that contain an application file name, command line parameters, and the current directory. The other parameters are security, environmental, and process information. The LoadLibrary() function takes one parameter that is a pointer to a string that contains the application file name:

```
HINSTANCE LoadLibrary(// Prototype from Microsoft Visual C++ Help Documentation
  LPCTSTR lpLibFileName        // address of filename of executable module
 );
```

The LoadLibraryEx() function takes three parameters the first being the same as LoadLibrary(), the second parameter must be null, and the third tells the operating system whether to load the file as an executable or as a data file in order to retrieve resources such as icons or string table data from it and not load it as an executable:

```
HINSTANCE LoadLibraryEx(// Prototype from Microsoft Visual C++ Help Documentation
  LPCTSTR lpLibFileName,       // points to name of executable module
  HANDLE hFile,                // reserved, must be NULL
  DWORD dwFlags                // entry-point execution flag
 );
```

The CreateFile() function is used to create and open files and to load files such as device drivers. This function also requires a pointer to a string that contains the name of a physical file:

```
HANDLE CreateFile(// Prototype from Microsoft Visual C++ Help Documentation
  LPCTSTR lpFileName,                              // pointer to name of the file
  DWORD dwDesiredAccess,                           // access (read-write) mode
  DWORD dwShareMode,                               // share mode
  LPSECURITY_ATTRIBUTES lpSecurityAttributes,      // pointer to security descriptor
  DWORD dwCreationDistribution,                    // how to create
  DWORD dwFlagsAndAttributes,                      // file attributes
  HANDLE hTemplateFile                             // handle to file with attributes to copy
 );
```

There are other functions such as MapViewOfFile() and MapViewOfFileEx() that map areas of memory to an already opened physical file through a handle to that file. They have the following parameters:

```
LPVOID MapViewOfFile(// Prototype from Microsoft Visual C++ Help Documentation
  HANDLE hFileMappingObject,        // file-mapping object to map into address space
  DWORD dwDesiredAccess,            // access mode
  DWORD dwFileOffsetHigh,           // high-order 32 bits of file offset
  DWORD dwFileOffsetLow,            // low-order 32 bits of file offset
  DWORD dwNumberOfBytesToMap        // number of bytes to map
 );
```

```
LPVOID MapViewOfFileEx(// Prototype from Microsoft Visual C++ Help Documentation
  HANDLE hFileMappingObject,        // file-mapping object to map into address space
  DWORD dwDesiredAccess,            // access mode
  DWORD dwFileOffsetHigh,           // high-order 32 bits of file offset
  DWORD dwFileOffsetLow,            // low-order 32 bits of file offset
  DWORD dwNumberOfBytesToMap,       // number of bytes to map
  LPVOID lpBaseAddress              // suggested starting address for mapped view
 );
```

All of the foregoing functions directly use a pointer to a string that is a physical file. The only file functions that do not directly use a physical filename are functions like CreateNamedPipe(), which has the following parameters:

```
HANDLE CreateNamedPipe(// Prototype from Microsoft Visual C++ Help Documentation
  LPCTSTR lpName,                                  // pointer to pipe name
  DWORD dwOpenMode,                                // pipe open mode
  DWORD dwPipeMode,                                // pipe-specific modes
  DWORD nMaxInstances,                             // maximum number of instances
  DWORD nOutBufferSize,                            // output buffer size, in bytes
  DWORD nInBufferSize,                             // input buffer size, in bytes
  DWORD nDefaultTimeOut,                           // time-out time, in milliseconds
  LPSECURITY_ATTRIBUTES lpSecurityAttributes       // pointer to security attributes structure
 );
```

The string to which CreateNamedPipe() points using the first parameter is a string that both an existing executable and the operating system know about and does not exist physically. Unfortunately both of the executables that "know" this private name could only be loaded using one of the other procedures that required a physical file. Currently it is not possible to load an executable using a "named pipe" name. Both of or any executables that use the name of the "named pipe" already must have been loaded into memory.

All of the foregoing functions require a physical file because all of them use "file mapping" processes. File mapping allows large executable files to appear to be loaded rapidly since they are rarely completely loaded into memory but rather are mapped into memory. The detriment to this mapping capability is that executable code must remain in physical memory in a file in unencrypted form in order to be loaded, unless there is a middle layer or file system driver that the operating system uses as a physical layer and that decrypts the executable code to the operating system on demand. The potential weakness here is that another file system driver can hook into the operating system to monitor traffic between the operating system and all file system drivers and capture decrypted executable code passing from the file system driver to the operating system. Some operating systems allow such monitoring more than others. Many anti-viral software packages use this technique to prevent computer virus attacks.

One method of loading and executing encrypted executable computer program code is to use a stub executable having two parts. The first part is the normal front end loader code that all executables have. In addition, the first part would perform any authorization which may include receiving a password from the user, then allocate enough memory to hold hidden encrypted code when it is decrypted, either in its entirety or a portion of it, copy the encrypted code into that area of protected (and preferably locked so no disk swapping occurs) memory, decrypt it once it is in memory and only in memory, and then have the operating system load the code only from memory therefore bypassing any file system drivers or TSRs so they have access to only encrypted code.

Some of the file functions listed above and similar functions on other operating systems could be modified easily by a programmer having access to source code for those operating systems, or a new operating system may be made to provide functions which allow direct loading of executable code from memory rather than physical files. For example, in the Win32 commands, a command similar to CreateProcess() command could be provided. The command should have a few extra parameters including the process identifier of the process that contains the now decrypted executable code, the memory address of the start of the decrypted code, and the size of the decrypted code. The command could also contain a parameter specifying a "call back" function within the first process that would provide decrypted code on demand directly to the operating system through a protected buffer, therefore allowing only a portion of the encrypted code to be decrypted at any one time instead of in its entirety, for better protection and less memory use. The second parameter of the LoadLibraryEx() command that now needs to be NULL could be expanded to hold a structure that contained the same information. Both of these and other similar functions could be changed or created to allow loading executable code either as an .exe, .dll, or other extensions or identifiers, such as by using a "named pipe" name that only the operating system and process that holds decrypted code know about and having the operating system load from the named pipe.

Alternatively, without having such additional capabilities in the operating system, an application program can be divided into two parts. The first part is code that is common to all applications such as code for allocating memory off the heap and code that provides some interaction with the user. This kind of code is generally not code that the content provider is concerned about copying. The second part is the code that the content provider believes is valuable. Typically this valuable code is a business logic code or what would be considered a middle tier of a three-tier environment. A content provider would like to protect this second part of the code, at least much more that the first part of the code. The content provider would place all of the important code to be protected inside a dynamic link library and the code that is not that important would reside in the front end "stub" executable. Both of these would be combined into another executable containing the .dll in encrypted form only, along with any other files, data, information, and/or tables for holding, for example, hardware identifiers. This other executable is the final digital information product.

The first part of the digital information product, i.e., the executable stub, would load and execute normally like any other application. It then would perform any authorization procedures. Once the proper authorization or password was completed successfully, an unwrap procedure would be performed as will now be described in connection with Fig. 8, it would then allocate enough protected memory using a function like VirtualAlloc() as shown in step 150:

```
DWORD nFileSize = 0;
DWORD nPhantomFileSize = 0;
```

```
DWORD exeOffset = 0;
DWORD nPreferredLoadAddress = GetPreCompiledLoadAddress();
CString cCommandFile = UnwrapGetNTCommandFile();
exeOffset = UnwrapGetDllOffset(cCommandFile);
nFileSize = UnwrapGetDllSize(cCommandFile);
nPhantomFileSize = nFileSize + 0x3000; // add any needed extra space
// Increase buffer size to account for page size (currently Intel page size).
DWORD nPageSize = GetPageSize();
nPhantomFileSize += (nPageSize -(nPhantomFileSize % nPageSize));
// Allocate the memory to hold the decrypted executable.
LPVOID lpvBlock = VirtualAlloc((LPVOID) nPreferredLoadAddress,
                    nPhantomFileSize,
                          MEM_RESERVE | MEM_COMMIT, PAGE_READWRITE);
```

This function can request a particular address space. Preferably, this address space is the preferred load address space to which the .dll was linked in order to minimize any needed relocation and fix up code. The stub executable may lock that area of memory in step 152, for example by using VirtualLock() to prevent any memory writes to a swap file, depending on the operating system, as shown below:

```
BOOL bVLock = VirtualLock((LPVOID) nPreferredLoadAddress, nPhantomFileSize);
```

The memory area still should be secure even without this preventive step since the Windows 95 and NT operating systems do not allow any user access to swap files.

The encrypted code is then copied from the digital information product into the allocated protected memory in step 154, for example by using the following command:

```
UnwrapCopyHiddenExeToMem(cCommandFile, exeOffset, nFileSize, (char *) lpvBlock);
```

Once in memory, the stub would then decrypt the code to that same portion of memory in step 156, for example by using the following commands:

```
CwrapDecryptSeed(cPassword.GetBuffer(0), cPassword.GetLength());
CwrapDecrypt((unsigned char *) lpvBlock, 0, nFileSize);
```

Any "fix up and relocation" type services would then be performed in step 158, for example by using the following command:

```
UnwrapFixUpAndRelocateDll(lpvBlock);
```

Possibly, the memory protection may be changed to execute only in step 160, for example by using the VirtualProtect() command as follows:

```
DWORD lpflOldProtect; // variable to get old protection
BOOL bVProtect = VirtualProtect((LPVOID) nPreferredLoadAddress,
                                nPhantomFileSize,
                                PAGE_EXECUTE,
                                &lpflOldProtect);
```

Function calls then can be made into that area of memory that now contains the decrypted code:

```
UnwrapDoDllAlgorithms();
```

Some of the "fix up" operations to be performed above include placing the addresses of external or stub.exe functions into the address place holders of the decrypted .dll or internal code, by using commands similar to the following:

```
WriteAddress((char*) 0x0a406104, (DWORD) &CallBackFunction1);
WriteAddress((char*) 0x0a406100, (DWORD) &CallBackFunction2);
```

For instance a wrapper function could be created in the outer stub.exe that received a size parameter, allocated that amount of memory off of the heap, and passed back the starting address of that block of memory. Another example would be to have encrypted algorithms within the hidden, encrypted .dll which would be called at run time from the front end stub once decrypted within protected memory. The dynamic link library would be compiled and linked to expect a pointer to a function that took that parameter and/or returned a value by including prototypes in the header file as follows:

```
void (*lpCallBackFunc1)();
void (*lpCallBackFunc2)(unsigned long);
```

Function calls to "external" functions also could be added as follows:

```
(*lpCallBackFunc1)();
unsigned long z = x * x;
(*lpCallBackFunc2)(z);
```

At run time the "fix up" code would take the run time address of that "wrapper function" and place it into the pointer address within the .dll block of code as follows:

```
WriteAddress((char*) 0x0a406104, (DWORD) &CallBackFunction1);
WriteAddress((char*) 0x0a406100, (DWORD) &CallBackFunction2);
```

This information is readily available using the .cod output files from the compiler, an example of which follows:

```
_TestSum PROC NEAR                                  ; COMDAT
; Line 8
 00000      56            push   esi
; Line 23
 00001      ff 15 00 00 00
        00          call   DWORD PTR_lpCallBackFunc1
; Line 24
 00007      8b 44 24 08   mov    eax, DWORD PTR_a$[esp]
 0000b      50            push   eax
 0000c e8 00 00 00 00 call   _TestSquare
```

```
 00011      83 c4 04        add    esp, 4
 00014      8b f0           mov    esi, eax
; Line 25
 00016      8b 44 24 0c     mov    eax, DWORD PTR_b$[esp]
 0001a      50              push   eax
 0001b      e8 00 00 00 00  call   _TestSquare
 00020      83 c4 04        add    esp, 4
 00023      03 c6           add    eax, esi
; Line 28
 00025      5e              pop    esi
 00026      c3              ret    0
_TestSum ENDP
_TEXT       ENDS
;       COMDAT_TestSquare
_TEXT       SEGMENT
_x$ = 8
_TestSquare PROC NEAR                                   ; COMDAT
; Line 30
 00000      56              push   esi
; Line 32
 00001      8b 74 24 08     mov    esi, DWORD PTR _x$[esp]
 00005      0f af f6        imul   esi, esi
; Line 34
 00008      56              push   esi
 00009      ff 15 00 00 00
         00          call   DWORD PTR _lpCallBackFunc2
 0000f 83 c4 04      add    esp, 4
 00012      8b c6           mov    eax, esi
; Line 36
 00014      5e              pop    esi
 00015      c3              ret    0
_TestSquare ENDP
```

Such information also is available from .map output files from the linker where the "f" between the address (i.e., 0a406100) and the object file (i.e. Algorithms.obj) means it is a "flat" address (i.e., hard coded by the linker) and the lack of an "f" means that it is an address pointer to be supplied at run time (load time) where the address that is contained in that address location is used and not the actual address location (i.e., the address that is contained at address location 0a406100 and not 0a406100 itself):

```
0001:00000000    _TestSum             0a401000 f Algorithms.obj
0001:00000030    _TestSquare          0a401030 f Algorithms.obj

0003:00001100    _lpCallBackFunc2     0a406100   Algorithms.obj
0003:00001104    _lpCallBackFunc1     0a406104   Algorithms.obj
```

When the code inside the .dll makes a "call" to a dereferenced pointer, it would jump to the correct function in the outer code and return the expected return value (if any). For example:

```
void CallBackFunction1(){
// This is the first function that exists in the Stub executable
// whose address has been placed at the appropriate location inside the "dll" code
// that has now been decrypted in a block of memory. The code inside the "dll"
// makes a function call to this function. In its encrypted state, the "dll" does not contain
// this address, but merely has a placeholder for the address. The "dll" has enough space
allocated to hold an
// address of this size. After the "dll" has been decrypted at run time, its address is
// placed in that location so the code inside the "dll" that references (or more
// appropriately dereferences) that address can jump (which is function call) to this
// address.
AfxMessageBox(
_T("This is the FIRST Stub.exe call back function being called from the dll."));
        return;
}
```

```
void CallBackFunction2(DWORD nNumber){
// See comment for CallBackFunction1 except this function receives a parameter off
// of the stack. It could also return a value as well.
        CString
        cString(
T("This is the SECOND Stub.exe call back function being called from the dll"));

        har buffer[20];
        ltoa(nNumber, buffer, 10);

        cString += _T(" with a parameter of ");
        cString += buffer;
        cString += _T(".");
        AfxMessageBox(cString.GetBuffer(0));
        return;
}
```

The outer stub.exe would make the same kinds of jumps or function calls into the now protected decrypted code block as follows:

```
DWORD c;

// This command declares a function pointer. This command is different for different function
// calls. Here the called function takes two integer parameters and
// passes back a DWORD.
DWORD (*lpFunc)(DWORD,DWORD);

// The function pointer is then pointed to the starting address of the function in the
// block of memory that now holds the decrypted DLL.
lpFunc = (DWORD (*)(DWORD,DWORD)) UnwrapFixUpAndRelocateDll();

// Now call that "function" which is really like all function calls, i.e., a jump to
```

```
// the address where that function exists. In this case, two
// variables are passed to that function and returning a value from that function. This function illustrates that the function call
// can be more complicated than merely a simple jump
// to an address. Inline assembler code may be used to push the variables onto
// the stack frame and return the variable from the eax register, but this function enables
// the C++ compiler to do the same function.
c = (DWORD) (*lpFunc)(a, b);
```

This mechanism requires the unwrap procedure and the now decrypted code to have intimate knowledge about procedural interfaces of each other but no knowledge about each other's implementation. This is the way most executable .exe files and .dll files behave but with the addition of a series of "wrapper" functions on either side for communication. This method works under Windows 95 and Windows NT 4.0 operating systems and should work under Windows NT 3.51 and other operating systems.

Another modified version of this mechanism that works under the Windows NT 4.0 operating system because of functions specific to Windows NT 4.0 would be to have another hidden and/or encrypted executable within the digital information product. This executable would be copied to a physical disk in an unencrypted form, launched or loaded with the CreateProcess() command in its current form but called with a parameter to load the executable in suspended mode:

```
BOOL success = CreateProcess(cFrontEndExe.GetBuffer(0), 0, 0, 0, TRUE,
                    CREATE_NEW_CONSOLE | CREATE_SUSPENDED,
                 0, 0, &startUpInfo, &processInfo);
```

Then the first process would copy the encrypted dll into its own process and decrypt it, allocate enough memory using VirtualAllocEx() in its current form in the second process that has just loaded the expendable front end executable in a suspended state as follows:

```
LPVOID lpvBlockEx = VirtualAllocEx(processInfo.hProcess,
```

```
                    (LPVOID) nPreferredLoadAddress,  nPhantomFileSize,
                        MEM_RESERVE | MEM_COMMIT,
                            PAGE_READWRITE);
```

The decrypted code is copied from the first process to the second suspended process using WriteProcessMemory() in its current form:

```
BOOL bWriteProcessMemory = WriteProcessMemory((HANDLE) processInfo.hProcess,
                (LPVOID) lpvBlockEx, (LPVOID) nPreferredAddress,
            (DWORD) nPhantomFileSize,  (LPDWORD) &nBytesWritten);
```

The primary thread of the previously launched second process is then resumed:

```
DWORD nResumed = ResumeThread(processInfo.hThread);
```

Any necessary function pointers are then placed in the correct locations by the second process, the area of memory is locked to prevent any writes to a swap file, and the memory protection is changed to execute only as follows:

```
WriteAddress((char*) 0x0a406104, (DWORD) &CallBackFunction1);
WriteAddress((char*) 0x0a406100, (DWORD) &CallBackFunction2);

BOOL bVLock = VirtualLock((LPVOID) nPreferredLoadAddress, nPhantomFileSize);
DWORD lpflOldProtect; // variable to get old protection
BOOL bVProtect = VirtualProtect((LPVOID) nPreferredLoadAddress,
                    nPhantomFileSize, PAGE_EXECUTE,     &lpflOldProtect);
```

The program can continue running by making and receiving calls to and from the decrypted dynamic link library that now resides in the protected memory of its process using commands such as the following:

```
DWORD c;
```

```
DWORD (*lpFunc)(DWORD,DWORD);
lpFunc = (DWORD (*)(DWORD,DWORD)) ExpendableGetEntryAddress();
c = (DWORD) (*lpFunc)(a, b);
```

The first process can either close down or launch another instance of that same process.

In either of these implementations using the same process or launching into a second process, the hidden encrypted code never passes through a file system driver or memory resident program in decrypted form. Code can be split up among different dynamic link libraries so that no two would reside in memory at the same time in order to protect code further. Both of these systems can be implemented using the Win32 function calls. If additional functions, similar to a CreateProcess() command or a LoadLibrary() command but that take a process identifier and address location in memory to load in an executable instead of a physical file, are provided in an operating system then the entire executable and dynamic link library can be hidden, encrypted, and protected on the physical disk and then decrypted within protected memory and use the operating system loader to load it directly to the operating system from memory without residing in decrypted form on any physical medium.

Having described the operation and use of the computer program product in accordance with the invention, embodiments of which are described above in connection with Figs. 3-8, and the operation of the unwrap procedure and device driver it contains, the process of constructing such a computer program product will now be described in more detail. Referring now to Fig. 9, an embodiment of this process for creating a computer program product is shown. This process can be applied to any digital information including an arbitrary executable computer program, dynamic link libraries and related files of data. All digital information is treated as mere data by this process. Each separate data file is combined into a single file by this process, with an executable program for performing the unwrap procedure, and optionally executable program code for a virtual device driver, into the computer program product. Each file of hidden information has a unique location and is identified by its own begin and end markers as shown in Fig. 3. The first step of this process is opening a new data file for the computer program using a name that will be used to indicate an executable file (step 200). For example, an executable word processing program may be named "word_processor.exe" in the Windows95 operating system.

The three portions of the computer program product are then inserted into the open data file. First, the unwrap procedure is inserted at the beginning of the file in an executable format in step 202. The begin tag for the optional device driver is then inserted in step 204. The executable device driver program code is then inserted in step 206, followed by its corresponding end tag in step 208. For each hidden file to be inserted into this computer program product, steps 210 to 216 are performed. First, the begin tag is inserted in step 210. The begin tag also may include an indication of a unique name of the file which will be used as its name in the phantom directory created by the unwrap procedure. The hidden file is then encrypted and/or compressed in step 212 and inserted into the data file in step 214. The end tag for the hidden file is then inserted in step 216. The device driver and all of the tags may be encrypted also if the unwrap procedure has suitable decryption procedures. The computer program file is closed when the last hidden file is processed.

Using the present invention digital information, such as executable program code or various kinds of data, is loaded and unloaded as needed, and thus does not take up any more memory than is necessary. At no time does unencrypted digital information, such as computer program code, exist on disk in accessible and complete decrypted form. Because the original digital information is available as a read only file in one embodiment of the invention accessible only to the device driver, the digital information may be accessed over networks, from a CD-ROM or from a DVD, and can be made to have a limited number of uses. This mechanism is particularly useful for controlling distribution of computer programs, digitized movies or other information while reducing the cost of such distribution and control. For example, software may be distributed over a network on a single use basis, and charges may be levied on a per use basis. The ability to reverse engineer an application program also may be reduced.

One benefit with this system over some other systems for preventing unauthorized access to digital information is that the content provider maintains control of the encryption applied to the information how it may be decrypted. Any need for either a centralized facility or a predetermined decryption program is eliminated. An operating systems manufacturer or other platform vendor merely provides the capability for the information to be accessed and decrypted on the fly. Since the valuable information and any other tables of authorization codes, passwords, or hardware identifiers that the content provider may use to secure the information resides in one large encrypted file, it becomes difficult, if not impossible, for someone to determine just where any of this information exists.

A potential scenario with authorization procedure in which the present invention may be used is the following. A consumer purchases a DVD disk containing a movie. The user puts the disk into the player. This is the first time the disk is installed. The content provider's functions are loaded into the DVD chip, which looks in the encrypted table and sees that this is the first time this disk is being played. The player then displays on a screen a numeric identifier and toll free phone number. The consumer calls the toll free phone number and inputs the numeric identifier that was displayed on the screen. The content provider provides a numeric password based on the numeric identifier that the user inputs into the DVD. The content provider may develop a database of information about its consumers that also may be used to detect pirating of the digital information product. Now that this authorization has taken place, the software that the content provider wrote, and is now in the DVD chip, takes a hardware identifier from the DVD and encrypts it and puts it in the encrypted and buried table on the disk. Alteratively, the data may be decrypted in memory and re-encrypted back onto the disk using the hardware identifier as part of a key. Now that disk will run and show the movie and will only run on that DVD and no other. The content provider could allow for a table of hardware id's so they could limit the number of DVD's that disk would run on or a limited number of times it can be shown. It should be understood that many other authorization procedures may be used.

In the foregoing scenario, the movie is encrypted on the same disk inside of the encrypted file that contains the table and functions the content provider distributed. The movie is decrypted by the decryption functions contained in the file directly to the DVD chip. At no time does the movie reside anywhere in decrypted form. The content provider can protect the movie with any desired level of security (for both encryption and authorization).

In the present invention, the onus of protection of content does not reside with a hardware manufacturer or platform provider but in the hands of the content provider. The hardware manufacturer only provides the mechanism to protect the digital information through the operating system. The technique and implementation of protection resides in the hands of the content provider. This mechanism allows the content providers to change the level of security as needed without any modifications to the hardware. The security of the content is provided by the encryption/decryption algorithms, public/private keys, and authorization methods which are determined by the content provider. Even each individual product can have its own encryption/decryption algorithms and/or public/private keys. All of these can be changed and enhanced as the market demands.

The present invention also could be used for on-line or live use of digital information. For example, a movie could be retrieved on demand and recorded by a consumer. A set top box could receive the digital information, decrypt it, and then re-encrypt and store the information using, for example, a hardware identifier of the set top box. Since home movies digitally recorded would be encrypted using the hardware identifier of the device used in recording, that home movie could not be played on another or only on a limited number of other devices and/or for only a specified number of times depending on the wishes of the content provider. Since the algorithms are downloaded at the time of recording from a service provider, e.g., the cable company, the content provider (movie company) would provide the encrypted data to the service provider to present to their customers. The service provider need not be concerned with the encryption/decryption and authorization functions used by the content provider. Similar uses are possible with other data transmission systems including, but not limited to, telephone, cellular communications, audio transmission including communication and the like.

In another embodiment, the stub executable program is a first process that is implemented similar to a debugging tool such as the SoftIce debugger from NuMega Technologies or the WinDebug debugger from Microsoft Corporation for Ring 0 kernel level debugging for an Intel processor based architecture, or the CodeView debugger for ring 3 application level debugging. Such a debugger controls execution of a program to be debugged as a second process and steps through each program statement or opcode of the debugged program. The debugging tool could be modified to monitor each opcode that indicates a jump to a program fragment, such as each instruction or a block code. If the program fragment to be executed is not decrypted, the modified debugger decrypts the program fragment before the jump command is allowed to execute. Each program fragment may be re-encrypted after execution. Clearly, unnecessary debugging commands may be omitted from the modified debugger.

Having now described a few embodiments of the invention, it should be apparent to those skilled in the art that the foregoing is merely illustrative and not limiting, having been presented by way of example only. Numerous modifications and other embodiments are within the scope of one of ordinary skill in the art and are contemplated as falling within the scope of the invention as defined by the appended claims and equivalent thereto.

CLAIMS

1. A computer-implemented process for executing encrypted computer program logic while maintaining protection against copying of corresponding decrypted executable computer program logic, wherein the encrypted computer program logic is stored in association with first executable computer program logic, the process comprising the steps of:

through an operating system of a computer, reading, loading and executing the first executable computer program logic as a first process having a protected memory area defined by the operating system;

the first process decrypting the encrypted computer program logic into second executable computer program logic and storing the second executable computer program logic in the protected memory area; and

the first process causing loading and execution of the decrypted second computer program logic in the protected memory area.

2. The process of claim 1, wherein the encrypted computer program logic and the first executable computer program logic are stored in a single data file accessible through the operating system.

3. The process of claim 1, wherein the execution of the decrypted second computer program logic is performed as a second process having a second protected memory area defined by the operating system.

4. A digital information product including a computer readable medium having digital information stored thereon, the digital information including computer program logic defining first executable computer program logic, wherein the first executable computer program logic when executed performs the following steps:

storing the encrypted computer program logic in a data file accessible through an operating system of a computer, wherein the data file also includes first executable computer program logic;

through the operating system, reading, loading and executing the first executable computer program logic from the data file as a first process having a protected memory area;

the first process decrypting the encrypted computer program logic into second executable computer program logic and storing the second executable computer program logic in the protected memory area; and

the first process causing loading and execution of the decrypted second computer program logic in the protected memory area.

5. A computer system comprising:

a processor for executing computer program logic;

a main memory operatively connected to the processor for storing digital information including executable computer program logic at memory locations addressed by the processor; and

an operating system defined by executable computer program logic stored in the memory and executed by the processor and having a command which when executed by the processor defines means for creating a process in response to a request specifying a process identifier and a memory location in the main memory, wherein the process identifier indicates the process making the request and the memory location stores executable computer program logic which when executed defines the process.

6. A computer system having an operating system, for decrypting digital information, comprising:

means for storing the encrypted computer program logic in a data file accessible through the operating system, wherein the data file also includes first executable computer program logic;

means, invokable through the operating system, for reading, loading and executing the first executable computer program logic from the data file as a first process having a protected memory area;

the first process defining means for decrypting the encrypted computer program logic into second executable computer program logic and storing the second executable computer program logic in the protected memory area; and

the first process defining means for causing loading and execution of the decrypted second computer program logic in the protected memory area.

7. The computer system of claim 6, wherein the encrypted computer program logic and the first executable computer program logic are stored in a single data file accessible through the operating system.

8. The computer system of claim 6, wherein the execution of the decrypted second computer program logic is performed as a second process having a second protected memory area defined by the operating system.

9. A digital information product, including a computer readable medium with computer readable information stored thereon, wherein the computer readable information comprises:

a first portion of executable computer program logic; and

a second portion of encrypted digital information; and

wherein the first portion of executable program logic, when executed, defines means, operative in response to requests for digital information, for accessing the second portion of encrypted digital information, for decrypting the encrypted digital information, and for outputting the decrypted digital information.

10. The digital information product of claim 9, wherein the encrypted digital information is encrypted executable computer program logic.

11. A computer program product including a self-decrypting encrypted executable computer program, comprising:

a computer readable medium having computer program logic stored thereon, wherein the computer program logic defines:

a first module,

a second module,

wherein the first module, when executed by a computer, defines means for loading the second module into memory of the computer, and

a third module defining the encrypted executable computer program,

wherein the second module, when executed by a computer, defines means for communicating with an operating system of the computer to receive requests for program code from the encrypted executable computer program from the third module, and for processing the requests to access and decrypt the encrypted executable computer program and for providing the decrypted executable code from the third module to the operating system.

12. A process for executing encrypted executable computer programs on a computer system having a processor, memory and operating system, comprising the steps of:

receiving computer program logic having a first module defining a start up routine, a second module, and a third module containing the encrypted executable computer program;

executing the first module of the received computer program logic using the processor,

wherein the step of executing causes the second module to be loaded into the memory of the computer system, and

generating requests from the operating system for data from the encrypted executable computer program which are received by the second module, and

accessing and decrypting the encrypted executable computer program and returning the decrypted executable computer program to the operating system.

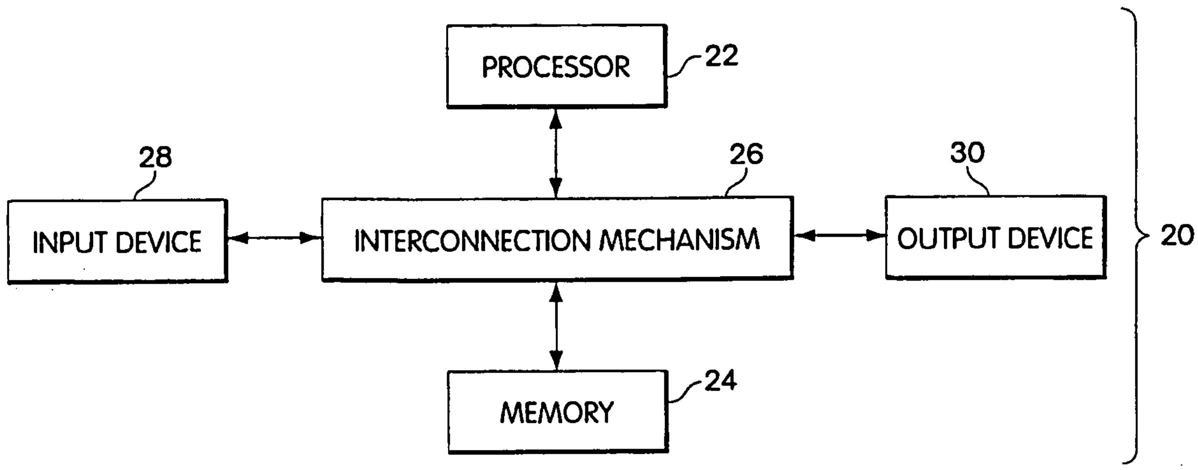

PROCESSOR
22
28
26
30
INPUT DEVICE
INTERCONNECTION MECHANISM
OUTPUT DEVICE
20
MEMORY
24

Fig. 1

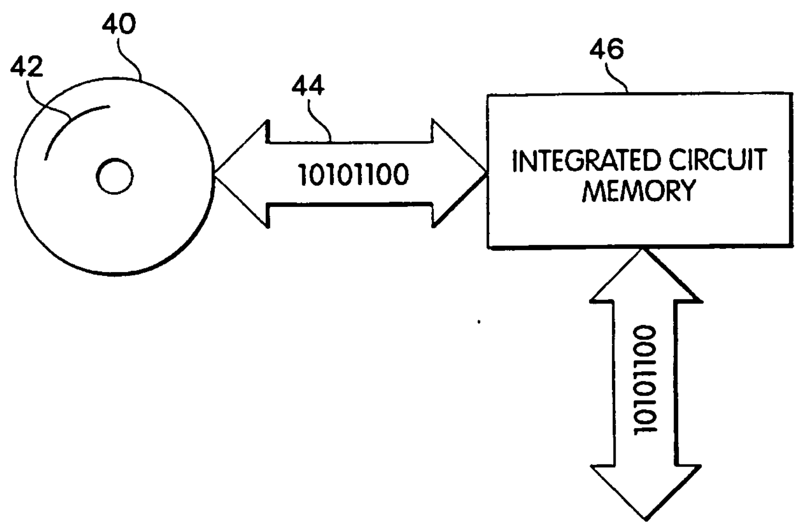

40
42
44
46
10101100
INTEGRATED CIRCUIT MEMORY
10101100
TO PROCESSOR 22

Fig. 2

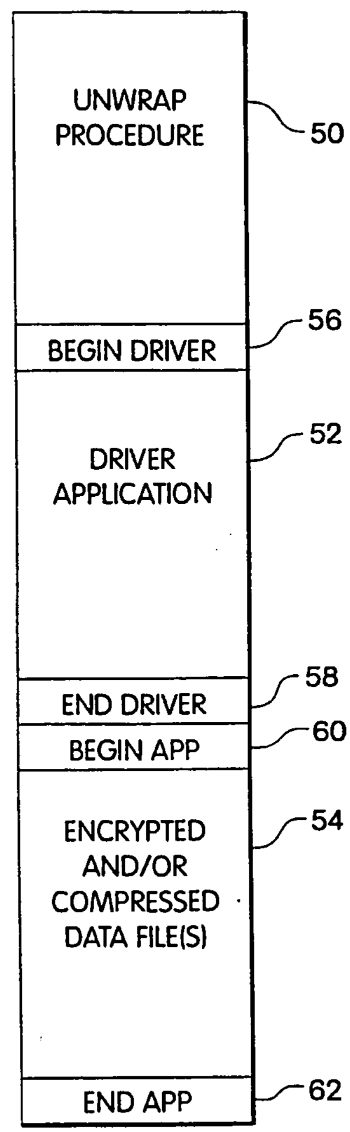

UNWRAP PROCEDURE
50
BEGIN DRIVER
56
DRIVER APPLICATION
52
END DRIVER
58
BEGIN APP
60
ENCRYPTED AND/OR COMPRESSED DATA FILE(S)
54
END APP
62

Fig. 3

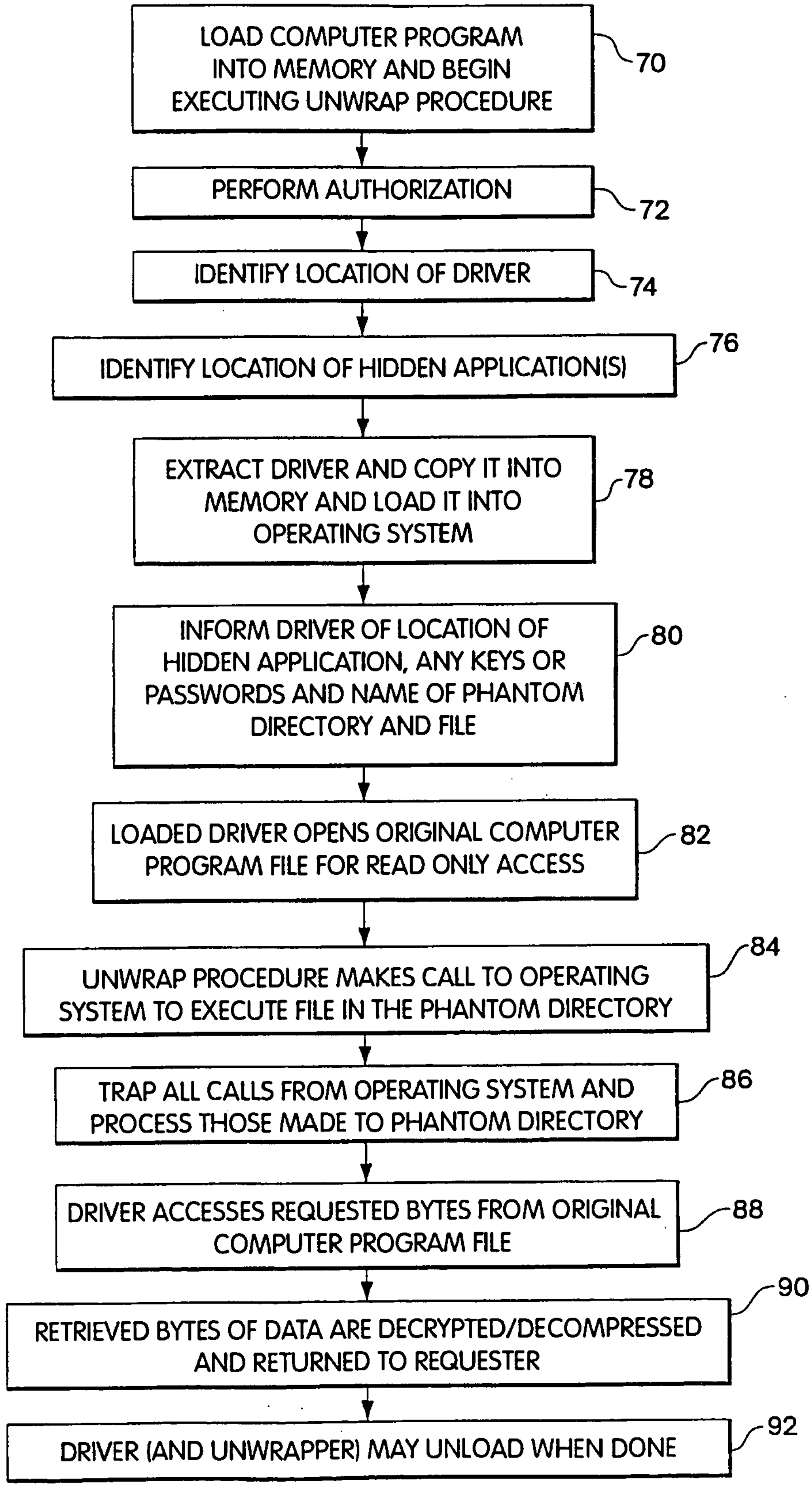

LOAD COMPUTER PROGRAM INTO MEMORY AND BEGIN EXECUTING UNWRAP PROCEDURE
70
PERFORM AUTHORIZATION
72
IDENTIFY LOCATION OF DRIVER
74
IDENTIFY LOCATION OF HIDDEN APPLICATION(S)
76
EXTRACT DRIVER AND COPY IT INTO MEMORY AND LOAD IT INTO OPERATING SYSTEM
78
INFORM DRIVER OF LOCATION OF HIDDEN APPLICATION, ANY KEYS OR PASSWORDS AND NAME OF PHANTOM DIRECTORY AND FILE
80
LOADED DRIVER OPENS ORIGINAL COMPUTER PROGRAM FILE FOR READ ONLY ACCESS
82
UNWRAP PROCEDURE MAKES CALL TO OPERATING SYSTEM TO EXECUTE FILE IN THE PHANTOM DIRECTORY
84
TRAP ALL CALLS FROM OPERATING SYSTEM AND PROCESS THOSE MADE TO PHANTOM DIRECTORY
86
DRIVER ACCESSES REQUESTED BYTES FROM ORIGINAL COMPUTER PROGRAM FILE
88
RETRIEVED BYTES OF DATA ARE DECRYPTED/DECOMPRESSED AND RETURNED TO REQUESTER
90
DRIVER (AND UNWRAPPER) MAY UNLOAD WHEN DONE
92

Fig.4

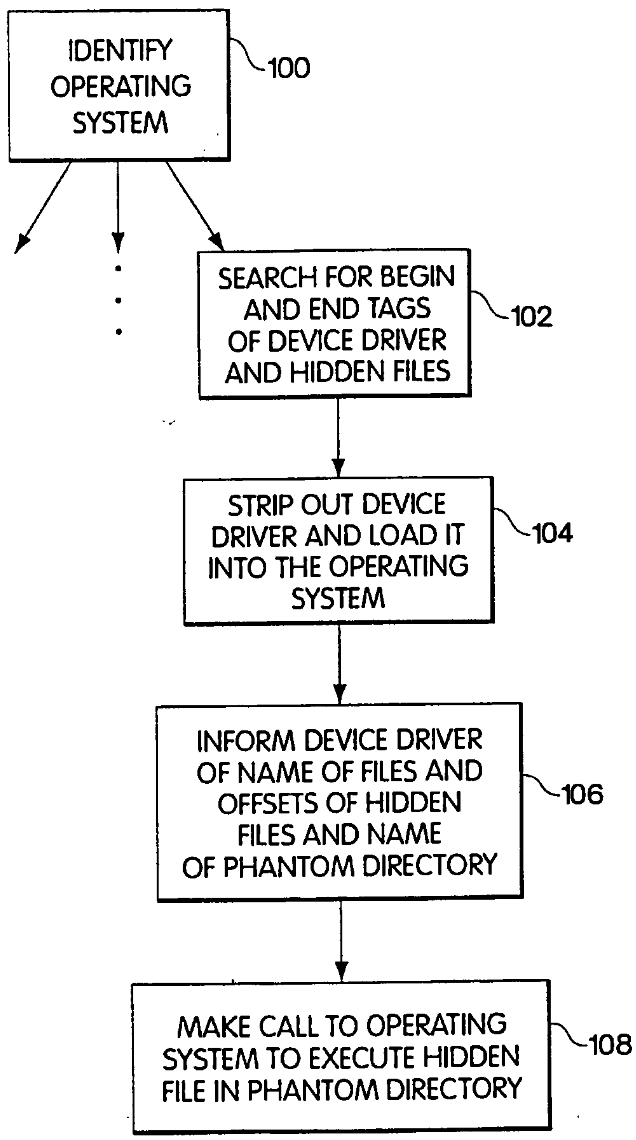

IDENTIFY OPERATING SYSTEM
100
SEARCH FOR BEGIN AND END TAGS OF DEVICE DRIVER AND HIDDEN FILES
102
STRIP OUT DEVICE DRIVER AND LOAD IT INTO THE OPERATING SYSTEM
104
INFORM DEVICE DRIVER OF NAME OF FILES AND OFFSETS OF HIDDEN FILES AND NAME OF PHANTOM DIRECTORY
106
MAKE CALL TO OPERATING SYSTEM TO EXECUTE HIDDEN FILE IN PHANTOM DIRECTORY
108

Fig. 5

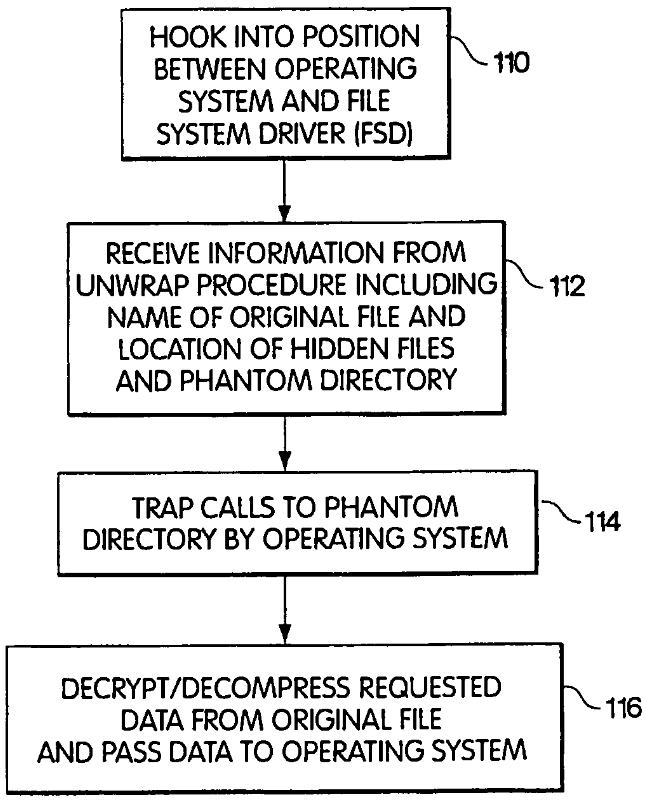

HOOK INTO POSITION BETWEEN OPERATING SYSTEM AND FILE SYSTEM DRIVER (FSD)
110
RECEIVE INFORMATION FROM UNWRAP PROCEDURE INCLUDING NAME OF ORIGINAL FILE AND LOCATION OF HIDDEN FILES AND PHANTOM DIRECTORY
112
TRAP CALLS TO PHANTOM DIRECTORY BY OPERATING SYSTEM
114
DECRYPT/DECOMPRESS REQUESTED DATA FROM ORIGINAL FILE AND PASS DATA TO OPERATING SYSTEM
116

Fig. 6

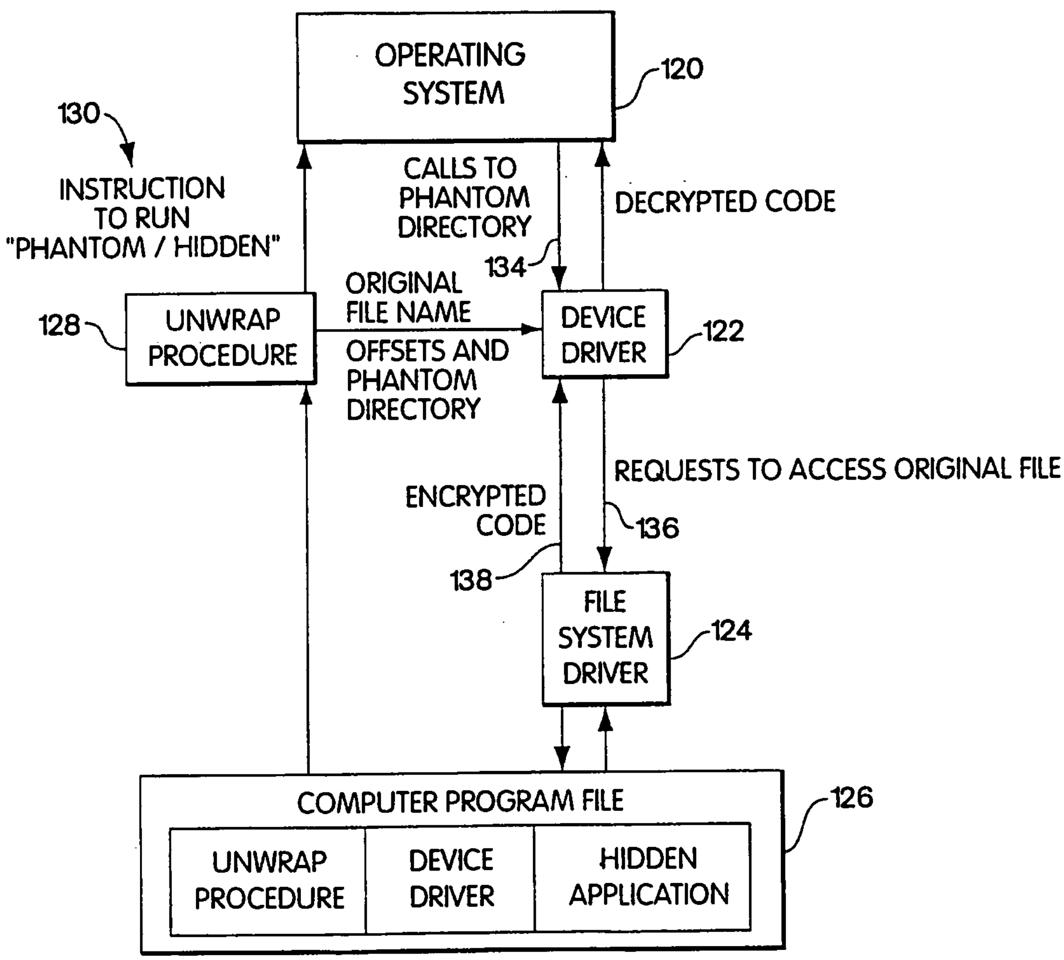

OPERATING SYSTEM
120
130
INSTRUCTION TO RUN "PHANTOM / HIDDEN"
CALLS TO PHANTOM DIRECTORY
DECRYPTED CODE
134
ORIGINAL FILE NAME
128
UNWRAP PROCEDURE
DEVICE DRIVER
122
OFFSETS AND PHANTOM DIRECTORY
REQUESTS TO ACCESS ORIGINAL FILE
ENCRYPTED CODE
136
138
FILE SYSTEM DRIVER
124
COMPUTER PROGRAM FILE
126
UNWRAP PROCEDURE
DEVICE DRIVER
HIDDEN APPLICATION

Fig. 7

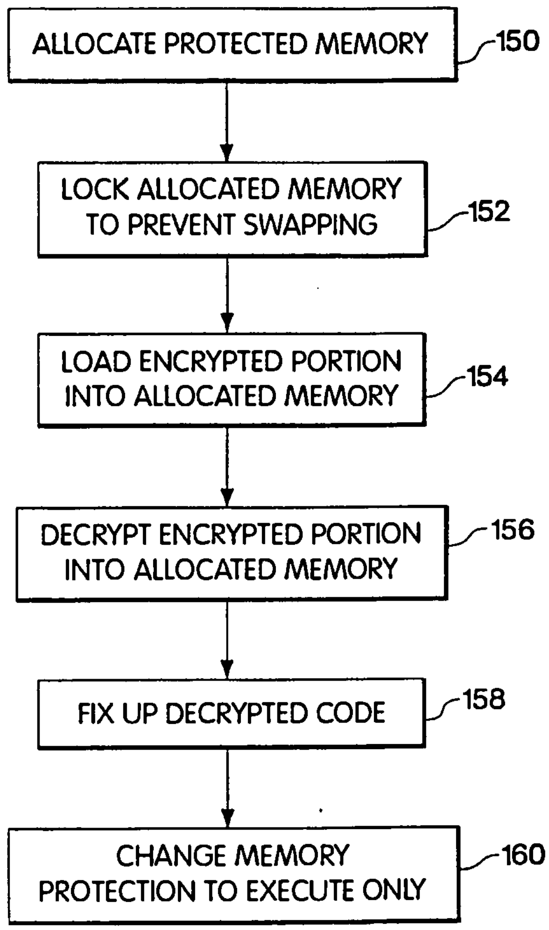

ALLOCATE PROTECTED MEMORY
150
LOCK ALLOCATED MEMORY TO PREVENT SWAPPING
152
LOAD ENCRYPTED PORTION INTO ALLOCATED MEMORY
154
DECRYPT ENCRYPTED PORTION INTO ALLOCATED MEMORY
156
FIX UP DECRYPTED CODE
158
CHANGE MEMORY PROTECTION TO EXECUTE ONLY
160

Fig. 8

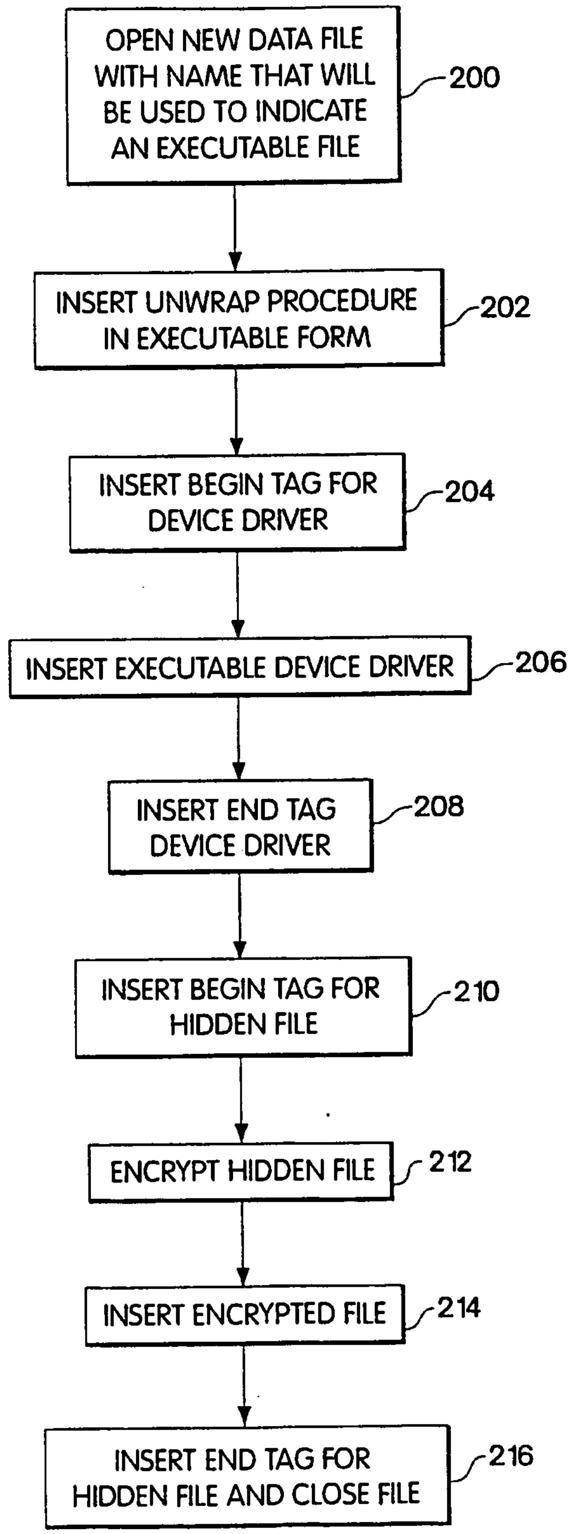

OPEN NEW DATA FILE WITH NAME THAT WILL BE USED TO INDICATE AN EXECUTABLE FILE
200
INSERT UNWRAP PROCEDURE IN EXECUTABLE FORM
202
INSERT BEGIN TAG FOR DEVICE DRIVER
204
INSERT EXECUTABLE DEVICE DRIVER
206
INSERT END TAG DEVICE DRIVER
208
INSERT BEGIN TAG FOR HIDDEN FILE
210
ENCRYPT HIDDEN FILE
212
INSERT ENCRYPTED FILE
214
INSERT END TAG FOR HIDDEN FILE AND CLOSE FILE
216

Fig. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/16223

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04L 9/00
US CL : 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. : 380/4,9,23,25,49,50,59

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)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 4,937,861 A (CUMMINS) 26 June 1990, see Abstract.	1-12
A	US 5,007,082 A (CUMMINS) 09 April 1991, see Abstract.	1-12
A	US 5,144,659 A (JONES) 01 September 1992, see Abstract.	1-12
A	US 5,155,827 A (GHERING) 13 October 1992, see Abstract.	1-12
A	US 5,396,609 A (SCHMIDT et al) 07 March 1995, see Abstract.	1-12

☐ Further documents are listed in the continuation of Box C. ☐ See patent family 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 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	Date of mailing of the international search report
20 JANUARY 1998	18 FEB 1998

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Diane Goodwin for BERNARR EARL GREGORY
Facsimile No. (703) 305-3230	Telephone No. (703) 306-4153

Form PCT/ISA/210 (second sheet)(July 1992)★

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : **H04L 9/08, H04Q 7/32**	A1	(11) International Publication Number: **WO 98/10561** (43) International Publication Date: 12 March 1998 (12.03.98)

(21) International Application Number: PCT/SE97/01407

(22) International Filing Date: 26 August 1997 (26.08.97)

(30) Priority Data:
08/708,796 9 September 1996 (09.09.96) US

(71) Applicant: TELEFONAKTIEBOLAGET LM ERICSSON (publ) [SE/SE]; S-126 25 Stockholm (SE).

(72) Inventor: RUNE, Johan; Motionsvägen 5, S-181 30 Lidingö (SE).

(74) Agents: BANDELIN, Hans et al.; Telefonaktiebolaget LM Ericsson, Patent and Trademark Dept., S-126 25 Stockholm (SE).

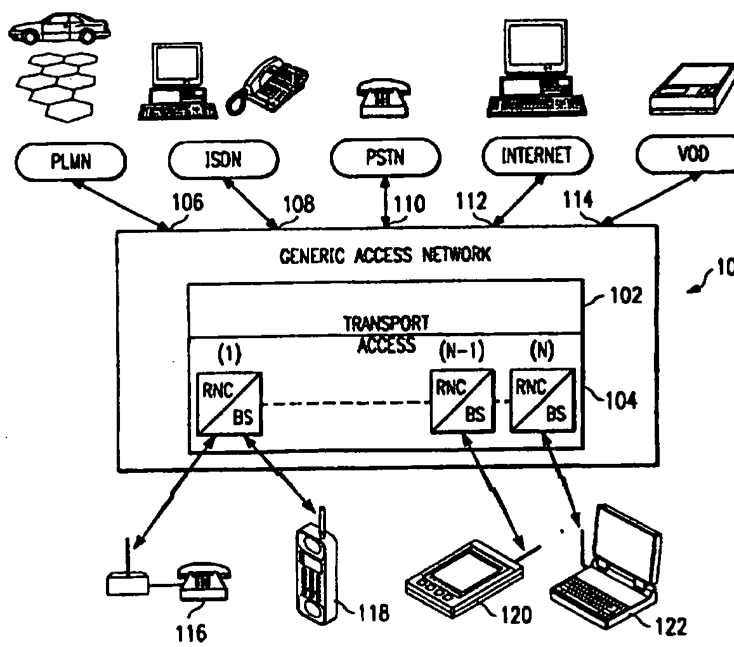
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, 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
With international search report.

(54) Title: METHOD AND APPARATUS FOR ENCRYPTING RADIO TRAFFIC IN A TELECOMMUNICATIONS NETWORK

(57) Abstract

A generic communications network (100) provides an encrypted communications interface between service networks (130, 132, 134) and their subscribers. When communications are initiated between a subscribing communications terminal (118) and the generic network (100), the terminal (118) compares a stored network identifier associated with a stored public key, with a unique identifier broadcast by the generic network (100). If a match is found, the terminal (118) generates a random secret key, encrypts the secret key with the stored public key, and transmits the encrypted secret key. The generic communications network (100) decrypts the secret key using a private key associated with the public key. The secret key is used thereafter by the terminal (118) and the generic network (100) to encrypt and decrypt the ensuing radio traffic. Consequently, the network (100) can maintain secure communications with the terminal (118) without ever knowing the terminal's identity.



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
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

METHOD AND APPARATUS FOR ENCRYPTING RADIO TRAFFIC IN A TELECOMMUNICATIONS NETWORK

BACKGROUND OF THE INVENTION

Technical Field of the Invention

The present invention relates generally to the field of wireless radio communications and, in particular, to a method and apparatus for encrypting radio traffic between terminals and a mobile communications network.

Description of Related Art

The need for increased mobility and versatility in telecommunications networks requires the networks to cover larger geographical areas and provide a broader range of telecommunications services to subscribers. These telecommunications services include teleservices and bearer services. A teleservice provides the necessary hardware and software for a subscriber to communicate with another subscriber (e.g., terminal, etc.). A bearer service provides the capacity required to transmit appropriate signals between two access points (e.g., ports) that provide an interface with a network. Telecommunications services can be provided to subscribers by a number of service networks, such as, for example, public land mobile telecommunications networks (PLMNs), public switched telephone networks (PSTNs), integrated services digital networks (ISDNs), the so-called "Internet" access networks, video on demand (VOD) networks, and other proprietary service networks.

In response to the need for increased mobility and versatility, a new mobile radio telecommunications network is being developed, which has a generic interface through which a service network subscriber can be connected with that service network regardless of the subscriber's geographic location. This generic mobile radio network is referred to as the "Generic Access Network" (GAN). In order to more readily understand the present invention, which deals primarily with encrypting communications traffic between terminals and a GAN, a brief description of such a GAN is provided below with respect to FIGURE 1.

FIGURE 1 is a perspective view of an exemplary GAN connected to a plurality of service networks and service network subscribers. The GAN (10) illustrated by FIGURE 1 includes an access network interconnected with a transport network. The access network includes a plurality of base stations (e.g., BS1 and BS2). Each base station includes a radio transmitter and receiver that provides communications coverage for a respective geographical area (e.g., a so-called cell, C1 and C2). The base stations are connected to a radio network controller (RNC) 12. Although not shown explicitly, certain of the base stations can be connected to RNC 12 (e.g., BS1 and BS2), and certain other of the base stations can be connected to one or more other RNCs. A plurality of the RNCs can be interconnected to provide a communications path therebetween. The RNCs distribute signals to and from the connected base stations.

A plurality of service networks (e.g., VOD network, PLMN, PSTN, Internet) are connected through respective access input ports (14, 16, 18, 20, 22, 24 and 26) to the access network of GAN 10. Each service network uses its own standard signaling protocol to communicate between its internal signaling nodes. For example, the Global System for Mobile communications (GSM), which is a digital cellular PLMN that has been fielded throughout Europe, uses the Multiple Application Part (MAP) signaling protocol. As illustrated by FIGURE 1, the RNCs in the access network are connected through at least one of the access input ports to a service network. As shown, RNC 12 is connected through access ports 20 and 24, respectively, to the PLMN and PSTN service networks.

Mobile terminals 28 and 30 are located within the radio coverage area of GAN 10, and can establish a connection with each of the base stations (e.g., BS2) in the access network. These mobile terminals can be, for example, a cellular phone, mobile radiotelephone, personal computer (notebook, laptop, etc.) possibly connected to a digital cellular phone, or mobile television receiver (for VOD). Signal transport between a mobile terminal and a selected service network takes place over specified signal carriers. For example, signals are transported between the cellular phone (28) and the PLMN service network over signal carriers SC1 and SC2.

The mobile terminals (e.g., 28 and 30) include an access section and service network section. The access section of a mobile terminal is a logical part of the access network and handles the signaling required to establish the signal carrier (e.g., SC2 and SC4) between the mobile terminals and RNC 12. The service network section of a mobile terminal is a logical part of the service network to which that terminal's user subscribes. The service network section of a mobile terminal receives and transmits signals, in accordance with the specified standards of its related service network, via the established signal carriers SC1 and SC2 (or SC4). The radio interface portion of the signal carrier SC2 or SC4 (between the mobile terminal and base station) can be time division multiple access (TDMA), code division multiple access (CDMA), or any other type of multiple access interface.

The service network subscribers can access their respective service network through the GAN. The GAN provides a signal carrier interface that allows a message to be transported transparently over a signal carrier (e.g., SC1 and SC2) between the service network part of a mobile terminal and its service network. The GAN accomplishes this function by matching the characteristics of the signaling connections and traffic connections of all of the service networks that connect to it. Consequently, the GAN can extend the coverage of existing service networks and also increase the subscribers' degree of mobility.

A unique characteristic of a GAN is that it has no subscribers of its own. The mobile users of the GAN are permanent subscribers to their own service networks, but they are only temporary users of the GAN. Consequently, a GAN does not know (or need to know) the identity of these users. However, a problem arises in attempting to encrypt radio traffic between the mobile terminals and the GAN.

Radio traffic (e.g., speech information or data) between mobile terminals and base stations is typically encrypted to ensure that the information being passed remains confidential. Although some service networks (e.g., GSM) encrypt traffic, most other service networks do not. Consequently, a GAN should be capable of encrypting traffic for those service networks that do not have that capability.

However, since a GAN does not know the identity of its users (the service network subscribers), it must be capable of encrypting radio traffic using encryption keys that are created without knowing a subscribing terminal's identity or authenticity. Unfortunately, most existing mobile communications networks use encryption techniques that generate encryption keys by using authentication parameters. In other words, to encrypt radio traffic in a conventional mobile communications network, the user terminal's identity must be known.

SUMMARY OF THE INVENTION

It is an object of the present invention to encrypt communications between a mobile terminal and a communications network without requiring the network to know the identity of the terminal.

It is also an object of the present invention to encrypt communications between a plurality of mobile terminals and a communications network without requiring the network to maintain individual encryption keys for each of the terminals.

It is another object of the present invention to encrypt communications between a mobile terminal and a communications network without requiring the terminal to permanently store a secret encryption key.

It is yet another object of the present invention to minimize call setup time, minimize transmission delays, and maximize data throughput, while encrypting communications between a mobile terminal and a communications network.

In accordance with one aspect of the present invention, a method is provided for encrypting communications between a communications network and a communications terminal, by storing a public key associated with the network at the terminal, generating a secret key at the terminal, encrypting the secret key with the stored public key at the terminal, transmitting the encrypted secret key from the terminal, receiving the encrypted secret key at the network, decrypting the received encrypted secret key with a private key, where the private key is associated with the public key, and encrypting the ensuing traffic with the secret key. If a public key has not been stored at the terminal, then the terminal transmits a request to the network for a public key. As such, the network is not required to know the identity of the terminal in order to maintain encrypted communications with the terminal.

In accordance with another aspect of the present invention, the foregoing and other objects are achieved by a method and an apparatus for encrypting traffic between a communications network and a communications terminal by broadcasting a (asymmetric) public key from the network. The public key is received by the terminal. The network maintains a private key that can be used to decrypt information encrypted with the public key. The terminal generates and stores a naturally occurring random number as a secret session (symmetric) key, encrypts the symmetric session key with the public key, and transmits the encrypted session key to the network. The network decrypts the session key with the private key, and both the network and terminal encrypt the ensuing communications with the secret session key. Again, the communications network is not required to know the identity of the terminal in order to maintain encrypted communications with the terminal.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the method and apparatus of the present invention may be had by reference to the following detailed description when taken in conjunction with the accompanying drawings wherein:

FIGURE 1 is a perspective view of an exemplary generic access network connected to a plurality of service networks and service network subscribers;

FIGURE 2 is a top level schematic block diagram of a generic access network in which a method of encrypting radio traffic between service networks and service network subscribers can be implemented, in accordance with a preferred embodiment of the present invention;

FIGURE 3 is a schematic block diagram of the access network illustrated in FIGURE 2;

FIGURE 4 is a sequence diagram that illustrates a method that can be used to encrypt radio communications between a generic access network and a terminal, in accordance with a preferred embodiment of the present invention; and

FIGURE 5 is a block diagram of a method that can be used to certify the authenticity of a public key and the owner of the key with a digital signature, in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE DRAWINGS

The preferred embodiment of the present invention and its advantages are best understood by referring to FIGUREs 1-5 of the drawings, like numerals being used for like and corresponding parts of the various drawings.

Essentially, in accordance with a preferred embodiment of the present invention, a mobile terminal stores at least one public key, along with a unique identification character of at least one GAN associated with that public key, in a memory location. A GAN continuously broadcasts its unique identification character in all cells connected to that GAN. When contact is initiated between the terminal and that GAN, the terminal compares the received identifier with the stored identifier(s), and if a match can be made, the terminal generates a random secret key, encrypts the secret key with the public key associated with that GAN's identifier, and transmits the encrypted secret key. The GAN decrypts the secret key using a private key associated with the public key. The secret key is used thereafter by the terminal and the GAN to encrypt and decrypt the ensuing radio traffic. Notably, the GAN can maintain secure communications with the terminal without ever knowing the terminal's identity. Furthermore, since the GAN does not need to know the identity of such a terminal, the GAN is not required to maintain a database of individual terminal encryption keys. Additionally, the terminal is not required to store its own secret key, because it can generate a new secret key for each communications session.

FIGURE 2 is a top level schematic block diagram of a generic access network in which a method of encrypting radio traffic between service networks and service network subscribers can be implemented, in accordance with a preferred embodiment of the present invention. A GAN 100 is shown, which includes a transport network 102 interconnected with an access network 104. A plurality of service networks (e.g., PLMN, ISDN, PSTN, INTERNET, VOD) are connected through respective access ports (e.g., 106, 108, 110, 112, 114) to transport network 102 and access network 104. Access network 104 includes a plurality of RNCs and associated base stations (e.g., RNC(1)-RNC(N)). The plurality of RNCs and associated base stations are connected by a respective radio interface to a plurality of mobile transceivers (terminals) 116, 118, 120 and 122. A user of each mobile terminal is a subscriber to at least one of the service networks PLMN, etc. The mobile terminals can communicate with their respective service networks in the manner described above with respect to FIGURE 1. More specifically, the RNCs control communications between the terminals and their respective service networks. Notably, although a plurality of mobile terminals (116, etc.) are shown in FIGURE 2, this is for illustrative purposes only. One or more fixed radio terminals may also be connected to GAN 100 and are thus capable of communicating with at least one of the service networks.

FIGURE 3 is a schematic block diagram of access network 104 illustrated in FIGURE 2. Access network 104 includes a plurality of RNCs (e.g., RNC(1)-RNC(N)). However, although a plurality of RNCs is shown for this embodiment, the present invention can be implemented with only one RNC. At least one service network (e.g., 130, 132, 134) is connected through at least one respective access port (e.g., AP1, AP(N-1), AP(N)) to at least one RNC. At least one base station (e.g., BS(1), BS(N)) is connected to a respective RNC (e.g., RNC(1), RNC(N)). Although a plurality of base stations is shown, the present invention can be implemented with only one base station.

A mobile terminal (e.g., cellular phone 118) is connected by a radio interface to base station BS(1). It should be readily understood that one terminal (118) is shown for illustrative purposes only and that one or more additional terminals could be shown. The RNCs (e.g., RNC(1)-RNC(N)) are interconnected by communications lines (136, 138) for communications therebetween. Consequently, terminal 118 can establish communications with any of the service networks (e.g., 130, 132, 134) through access network 104 and GAN 100 (FIGURE 2). Notably, the coverage provided for each service network can be enlarged by switching to a different access port of access network 104. In other words, terminal 118 can communicate with service network 132 through RNC(1), interconnecting line 136, and RNC(N-1). Alternatively, if service network 132 is switched to access port AP(1), terminal 118 can communicate with service network 132 through RNC(1).

FIGURE 4 is a sequence diagram that illustrates a method that can be used to encrypt radio communications between a generic access network and a terminal, in accordance with a preferred embodiment of the present invention. The method 200 of encrypting communications can begin at the GAN or the terminal. For example, in this embodiment, at step 204, the GAN (e.g., 10) continuously broadcasts a unique identification character in all cells connected to that GAN. The terminal (e.g., 118) contains a non-volatile memory located in a GAN section of the terminal. The terminal stores at least one public key in the non-volatile memory. Along with each public key, the terminal also stores a respective expiration date for the key, and a GAN identification character that identifies a specific GAN associated with that key. In other words, each public key stored in the terminal's memory is thereby associated with a specific GAN. The terminal initiates contact by registering with a GAN (but not necessarily setting up a call). A processor in the terminal compares the received GAN identifier with the stored identifiers, and if a match can be made (and the key has not expired), the processor retrieves the stored public key associated with the identified GAN. However, in the event that no such match is found, the terminal sends a request for the GAN to transmit a public key. The transmitted public key (and its expiration date) is stored in the terminal and can be used to encrypt a secret key in the current and ensuing communication sessions.

At step 206, the terminal generates a (symmetric) secret key (described in detail below). At step 208, the terminal uses the retrieved public key to encrypt the secret key. At step 210, the terminal transmits the encrypted secret key to the identified GAN. At step 212, the GAN decrypts the secret key, which, at step 214, is used by the GAN and the terminal for encrypting traffic during the ensuing communications session (described in detail below).

Alternatively, at the end of a session with a GAN, the terminal stores the public key used for that session. When the terminal or a GAN begins a new communications session, the terminal retrieves the public key stored from the last session with a GAN, and uses that public key to encrypt a secret key to be used for the ensuing session. If the use of that stored public key is unsuccessful, the terminal then sends a request to the GAN for a new public key. This technique advantageously increases network throughput, because a network channel is not tied up transmitting a public key. However, if a public key has not been stored from a past session with a particular GAN, the terminal can still receive the public key by requesting it from the GAN and using it to encrypt a secret key that will be used for the ensuing session. In any event, by storing the relatively large (bit-wise) public keys in the terminal, as opposed to transmitting them from the GAN, radio transmission delays can be reduced significantly, a substantial amount of network transmission time can be saved, and data throughput will be increased.

FIGURE 4 also illustrates a method that can be used to encrypt radio communications between a generic access network and a mobile terminal, in accordance with another embodiment of the present invention. For example, when communications are desired between a service network and a terminal (e.g., PLMN and terminal 118), the service network or terminal can initiate communications with a call setup message. At step 202, as the initial connection between the GAN and the terminal is established, the service network can request that the ensuing traffic will be encrypted. If so, at step 204, still during the initial call setup process, the terminal receives a public key which is continuously broadcast from one or more base stations (e.g., BS(1)-BS(N)).

In this embodiment, all of the RNCs maintain at least one public key/private key pair (the same pair in every RNC) in a memory storage location. The public key that was broadcast by the GAN is received by the terminal (118) that has initiated contact with that GAN. Preferably, both the call setup procedure and the procedure to transfer the public key is performed by an RNC, which is connected through an access port to the service network of interest (e.g., RNC(1) to AP(1) to PLMN 130). Alternatively, a base station (e.g., BS1) can be configured to maintain public/private key pairs and broadcast or otherwise transfer a public key to a terminal.

The RNC can broadcast the public key in all cells in the RNC's coverage area. Consequently, since the GAN broadcasts the public key instead of having the terminal request the key from the GAN, the terminal can register with the GAN much faster, and a call can be set up in a substantially shorter period of time. Alternatively, instead of broadcasting the public key in a plurality of cells, the RNC can transfer the public key directly through the base station that has established contact with the terminal. However, the method of broadcasting the public key in a plurality of cells before call setup advantageously decreases the load on the GAN's dedicated traffic channels.

For all embodiments, as long as the terminal is registered with the GAN, the same public key can be used for all subsequent communications with that GAN, because the same key is stored at the GAN and also at the terminal. Alternatively, the public key can be changed periodically in accordance with a predetermined scheme or algorithm, or even at the whim of the GAN operator. If an operator desires to change public keys periodically, storing each public key's expiration date at the terminal facilitates their use in that regard. Furthermore, in the preferred embodiment, when the public key is changed, it can be broadcast by the GAN for a predetermined period of time, to minimize the number of terminal requests for a new public key.

As described earlier, at step 202, the GAN can maintain one or more asymmetric public key/private key pairs. In that event, a so-called "RSA Algorithm" can be used to create the public key/private key pairs. The RSA Algorithm combines the difficulty of factoring a prime number with the ease of generating large prime numbers (using a probabilistic algorithm) to split an encryption key into a public part and a private part.

Specifically, assuming that the letters P and Q represent prime numbers, the letter M represents an unencrypted message, and the letter C represents the encrypted form of M, the RSA Algorithm can be expressed as follows:

$$M^E \bmod PQ => C \text{ (encrypted message M)} \qquad (1)$$

$$C^D \bmod PQ => M \text{ (decrypted message C)} \qquad (2)$$

where the term (DE-1) is a multiple of (P-1)(Q-1). In this embodiment, the exponent E is set to 3. The private and public keys are each composed of two numbers. For example, the numbers represented by (PQ, D) make up the private key, and the numbers represented by (PQ, E) make up the public key. Since the same value for E is used consistently, only the PQ portion of the number need be sent on request or broadcast and used for the public key (e.g., at step 204). By knowing the private key, any message encrypted with the public key can be decrypted.

Returning to FIGURE 4, at step 206, the terminal (118) receives and/or stores the asymmetric public key. The terminal generates a random symmetric secret key. The random secret key, which is used to encrypt communications preferably for the complete session, can be generated in at least one of four ways. Using one method, the terminal takes several samples from measurements of the strength of the received signal, concatenates the lower order bits of the several samples, and processes the result to produce a random number. Since the lower order bits of the received signal are well within the noise level of the received signal, a naturally occurring, truly random number is generated. A second random number generating method is to use the random noise signal created at the input of an A/D converter connected to a microphone. Again, using this method, a naturally occurring, truly random number can be generated for the secret key. A third random number generating method is for the terminal to take samples from phase measurements of the received signal, concatenate the lower order bits of the samples, and process the result to produce a random number. A fourth random number generating method is for the terminal to take samples from the encoding section of the speech codec, concatenate the lower order bits of the samples, and process the result to produce the random number.

Alternatively, a random number generated at the terminal can be used as a seed for a pseudorandom number generator. The seed is encrypted with the public key from the GAN, and transmitted to the GAN. The seed is used simultaneously in the GAN and the terminal to produce a pseudorandom number. The pseudorandom number thus generated can be used by the GAN and the terminal as the secret key for the ensuing communications session.

The session key can be changed periodically to a different number in the pseudorandom number sequence. For example, the session key can be changed for a number of reasons, such as after a predetermined amount of data has been encrypted, or after traffic has been encrypted for a predetermined amount of time. The terminal or the GAN can initiate a change of the secret key, or the key can be changed according to a predetermined scheme or algorithm. For example, a request to change the secret session key can be implemented by transmitting a "session key change request" message, or by setting a "session key change request" bit in the header of a transmitted message.

Additionally, shorter session keys can be generated and less complicated encryption algorithms can be used with the pseudorandom number generation method described above. Consequently, a substantial amount of processing power can be saved in the GAN and especially in the terminal. The terminal can be configured to select the length of the session key to be used, in order to address trade offs between security and computational requirements. For example, the terminal's processor can select the length of a secret session key by generating a session key at that length, or by specifying the number of bits to be used from the output of the pseudorandom number generator. Alternatively, the terminal can specify the range of the output of the pseudorandom number generator to set a predetermined length.

Other alternative methods may be used to generate a pseudorandom number for a secret session key. For example, using a "Lagged Fibonacci" type of pseudorandom number generator, the n^{th} number in the pseudorandom number sequence, N_n, can be calculated as follows:

$$N_n = (N_{n-k} - N_{n-l}) \bmod M \qquad (3)$$

where k and l are the so-called lags, and M defines the range of the pseudorandom numbers to be generated. For optimum results, the largest lag should be between 1000 and 10000. If a relatively long key is desired, a plurality of the pseudorandom numbers produced by equation 3 can be concatenated to produce a longer key. If the pseudorandom numbers produced by equation 3 are to be floating point numbers between 0 and 1, M can be set to 1. The bit patterns of such floating point pseudorandom numbers can be used as symmetric encryption keys.

Another pseudorandom number generator that can be used to create a secret session key is based on an algorithm that produces pseudorandom numbers uniformly distributed between 0 and 1. Specifically, the seeds X_0, Y_0 and Z_0 of the pseudorandom numbers N_n are initially set to integer values between 1 and 30000. The pseudorandom numbers N_n are then calculated as follows:

$$X_n = 171*(X_{n-1} \bmod 177)-(2*X_{n-1}/177) \qquad (4)$$

$$Y_n = 172*(Y_{n-1} \bmod 176)-(35*Y_{n-1}/176) \qquad (5)$$

$$Z_n = 170*(Z_{n-1} \bmod 178)-(63*Z_{n-1}/178) \qquad (6)$$

If any of the values of X_n, Y_n or Z_n are less than zero, respectively, then X_n is set equal to $X_n+30269$, Y_n is set equal to $Y_n+30307$, or Z_n is set equal to $Z_n+30323$. The pseudorandom numbers N_n are then equal to $((X_n/30269 + Y_n/30307 + Z_n/30323) \text{ amod } 1)$, where X_n, Y_n and Z_n are floating point numbers, and "amod" means that these numbers can be fractions. The floating point numbers generated with this algorithm form bit patterns that are suitable for use as symmetric encryption keys. The length of such keys can be extended by concatenating a plurality of the pseudorandom numbers generated.

Returning to the method illustrated by FIGURE 4, at step 208, preferably using the above-described RSA Algorithm, the terminal encrypts the secret symmetric key with the public key. For example, assume that the secret symmetric key generated at the terminal is represented by the letters SK. Using equation 1 of the RSA Algorithm, the secret key is encrypted as follows:

$$M^E \bmod PQ => C$$

where (PQ, E) represents the public key, M is equal to SK, and C is the encrypted version of SK. The exponent E is set to 3.

In the preferred embodiment, the terminal places the encrypted secret key into a message format, which includes a header and message field. The header provides control information associated with the encrypted secret key that follows in the message field. A bit in the header can be set to indicate that the message field that follows the header is encrypted. In other words, only the secret key field of the message is encrypted. The header of the message is transmitted in the clear. Consequently, a substantial amount of network processing time can be saved at the RNC, since the header indicates whether the subsequent message field is encrypted, and if so, only that portion of the message is to be decrypted.

At step 210, the terminal (118) transmits the encrypted secret key (C) to the GAN via the contacted base station (e.g., BS(1)). In the preferred embodiment, this secret key is used for the ensuing communications. Alternatively, at any time during the ensuing communications session, the terminal can generate a new secret key, encrypt it with the public key, and transmit the new encrypted secret key to the GAN. The security of the session is thereby increased, because by reducing the amount of time that a particular secret key is used for a session, the likelihood that the secret key will be broken by an unauthorized user is also reduced.

At step 212, the RNC (e.g., RNC(1)) receives the encrypted secret key (C) from the base station, and decrypts the secret key using the private key part of the RSA Algorithm. For example, using equation 2 (above) of the RSA Algorithm, the received encrypted secret key (C) is decrypted as follows:

$$C^D \bmod PQ => M$$

where (PQ, D) represents the private key, and M is equal to SK (secret key).

At step 214, the ensuing radio traffic between the RNC and the terminal is encrypted and decrypted with the secret key, which is now known to both the RNC and the terminal. A known symmetric encryption algorithm can be used to encrypt and decrypt the ensuing radio traffic with the secret key, such as, for example, a one, two or three pass Data Encryption Standard (DES) algorithm, or a Fast Encipherment Algorithm (FEAL).

As yet another encryption alternative, instead of using the RSA Algorithm to create a public/private key pair, a so-called Diffie-Hellman "exponential key exchange" algorithm can be used to let the terminal and the GAN agree on a secret session key. In using this encryption scheme, two numbers (α, q) are stored at the GAN. At the beginning of a communications session, the RNC transmits the two numbers directly (or broadcasts the numbers) to the terminal. The numbers α and q are required to meet the following criteria: q is a large prime number that defines the finite (Galios) field GF(q) = 1, 2..., q-1; and α is a fixed primitive element of GF(q). In other words, the exponents (x) of (α^x mod q) produce all of the elements 1,2..., q-1 of GF(q). In order to generate an agreed to secret session key, the two numbers (α, q) are transmitted directly (or broadcast) from the GAN to the terminal. Alternatively, the two numbers can be already resident in the terminal's non-volatile memory. The terminal (118) generates the random number $X_T(1<X_T<q-1)$, and computes the value of $Y_T=\alpha^{X_T} \bmod q$. The GAN (e.g., the RNC or base station) generates the random number $X_G(1<X_G<q-1)$, and computes the value of $Y_G=\alpha^{X_G}$ mod q. The random numbers can be generated at the terminal using the methods described above with respect to generating naturally occurring, truly random numbers.

Y_T and Y_G are transferred unencrypted to the respective GAN and terminal. Upon receipt of the number Y_G, the terminal calculates the value of $K_S=Y_G^{X_T} \bmod q=\alpha^{X_G X_T} \bmod q$. Upon receipt of the number Y_T, the GAN calculates the value of $K_S=Y_T^{X_G} \bmod q=\alpha^{X_T X_G} \bmod q$. The number X_T is kept secret at the terminal, the number X_G is kept secret at the GAN, but the value of K_S is now known at both the terminal and the GAN. The number K_S is therefore used by both as the communications session encryption key. An unauthorized user would not know either X_T or X_G and would have to compute the key K_S from Y_T and Y_G, which is a prohibitive computational process. A significant security advantage of using the exponential key exchange algorithm is that the GAN is not required to maintain secret private key data on a permanent basis.

In summary, when a communications session is first initiated between a GAN and a terminal, the terminal receives an asymmetric public key that has been continuously broadcast by the GAN, retrieved from the terminal's internal memory, or requested from the GAN. The GAN maintains a private key that can be used to decrypt information encrypted with the public key. The terminal generates and stores a naturally occurring random number as a secret session (symmetric) key, encrypts the symmetric session key with the public key, and transmits the encrypted session key to the GAN. The GAN decrypts the session key with the private key, and both the GAN and terminal encrypt the ensuing communications with the secret session key. A primary technical advantage of transferring a public key from a GAN to a terminal at the onset of communications is that the GAN is not required to know the identity of the terminal in order to have encrypted communications with the terminal. However, a problem can arise if an unauthorized user attempts to impersonate a GAN and transmits a public key to the terminal. In that event, as described below, the terminal can be configured to authenticate the received public key and the identity of the GAN.

For example, when a public key is to be transferred from a GAN to a terminal, the key can be transferred with a public key "certificate". This certificate provides proof that the associated public key and the owner of that key are authentic. A "trusted" third party can issue the public key along with the certificate, which includes a "digital signature" that authenticates the third party's identity and the public key. The certificate can also contain the GAN's identity and the expiration date of the certificate, if any.

In one aspect of the invention, the GAN transmits the certificate and public key to the terminal. In that case, the public key of the third party is pre-stored (a priori) at the subscribing terminals.

FIGURE 5 is a block diagram of a method that can be used to certify the authenticity of a public key and the owner of the key with a digital signature, in accordance with the present invention. The method (300) of digitally signing a public key certificate and verifying its authenticity begins at step 302. At step 302, a "certificate" containing unencrypted information about the owner of the public key to be transferred to a terminal is prepared by a trusted third party. The unencrypted information also includes the public key and the expiration date of the certificate. At step 304, the resulting "unsigned" certificate is processed with an irreversible algorithm (e.g., a hashing algorithm) to produce a message digest at step 306, which is a digested or shortened version of the information included on the certificate. At step 308, the digest information is encrypted with a private key of a different public/private key pair. Preferably, an RSA algorithm similar to equations 1 and 2 above is used to derive this key pair. At step 310, a digitally signed public key certificate is thereby produced that contains the originally unencrypted information (including the public key to be used for the communications session) and the digest information, which is now encrypted with the certificate issuer's private key. The digitally signed public key certificate is then transferred to the terminal that has initiated contact with the GAN.

At step 312, upon receiving the digitally signed certificate, the terminal's processor analyzes the unencrypted and encrypted portions of the document. At step 314, the unencrypted information is processed using an algorithm identical to the hashing algorithm used at step 304. At step 316, a second digested version of the unencrypted information is produced at the terminal. At step 318, the terminal's processor retrieves the pre-stored certificate issuer's public key from memory, and using an RSA algorithm, decrypts the encrypted digest information from the certificate. Another version of the unencrypted digested information is thereby produced at step 320. At step 322, the terminal compares the two versions of the unencrypted digested information, and if the compared information is identical, the certificate's signature and the session public key are assumed to be authentic. That certified public key can now be used by the terminal to encrypt the secret session key.

Although a preferred embodiment of the method and apparatus of the present invention has been illustrated in the accompanying Drawings and described in the foregoing Detailed Description, it will be understood that the invention is not limited to the embodiments disclosed, but is capable of numerous rearrangements, modifications and substitutions without departing from the spirit of the invention as set forth and defined by the following claims.

WHAT IS CLAIMED IS:

1. A method for encrypting communications traffic between a mobile communications network and a communications terminal, comprising the steps of:

storing a public key and a first identifier associated with said mobile communications network at said communications terminal;

comparing said first identifier stored at said communications terminal with a second identifier received from said mobile communications network and producing a first predetermined result;

generating a secret key at said communications terminal;

encrypting said secret key with said stored public key at said communications terminal; and

transmitting said encrypted secret key from said communications terminal.

2. The method according to Claim 1, further comprising the steps of:

receiving said encrypted secret key at said mobile communications network;

decrypting said received encrypted secret key with a private key, said private key associated with said public key; and

encrypting said communications traffic with said secret key.

3. The method according to Claim 1, wherein the step of storing a public key comprises the step of a priori pre-storing the public key.

4. The method according to Claim 1, further comprising the step of transmitting said public key from said mobile communications network upon receiving a public key request from said communications terminal.

5. The method according to Claim 4, wherein the step of transmitting said public key further comprises the step of transmitting information to authenticate said public key.

6. The method according to Claim 4, further comprising the step of transmitting said request from said communications terminal upon said comparing step producing a second predetermined result.

7. The method according to Claim 1, wherein the steps of receiving and decrypting said encrypted secret key are performed at a radio base station in said mobile communications network.

8. The method according to Claim 1, wherein the step of decrypting said received encrypted secret key is performed at a radio network controller in said mobile communications network.

9. The method according to Claim 1, wherein said mobile communications network comprises a generic communications network.

10. The method according to Claim 1, wherein said communications terminal comprises a mobile terminal.

11. The method according to Claim 1, wherein said communications terminal comprises a fixed terminal.

12. The method according to Claim 1, wherein said communications terminal comprises an unidentified communications terminal.

13. The method according to Claim 1, wherein said mobile communications network comprises a cellular phone network.

14. The method according to Claim 1, further comprising the steps of:

connecting a plurality of service networks to said mobile communications network, a user of said communications terminal being a subscriber to at least one of said plurality of service networks; and providing a communications path between said communications terminal and said at least one of said plurality of service networks.

15. The method according to Claim 1, wherein said private key and said public key are associated by an RSA Algorithm.

16. The method according to Claim 1, wherein said secret key comprises a symmetric encryption key.

17. The method according to Claim 1, wherein the step of generating a secret key comprises the step of generating a naturally occurring random number.

18. The method according to Claim 1, wherein the step of generating a secret key comprises the steps of:

detecting a received signal in digital form at said communications terminal; and

extracting at least one low order bit from said detected received signal.

19. The method according to Claim 1, wherein the step of generating a secret key comprises the steps of:

detecting a signal at an output of a microphone A/D converter; and

extracting at least one low order bit from said detected output signal.

20. The method according to Claim 1, wherein the step of generating a secret key comprises the steps of:

detecting a signal at an output of a speech codec; and

extracting at least one low order bit from said detected output signal.

21. The method according to Claim 1, wherein the step of generating a secret key comprises the steps of:

generating a seed for a pseudorandom number; and

generating a pseudorandom number from said seed.

22. The method according to Claim 1, wherein a length of said secret key is predetermined at said communications terminal.

23. The method according to Claim 1, wherein said secret key further comprises a plurality of concatenated numbers.

24. The method according to Claim 1, wherein the step of storing said public key and said first identifier further comprises storing an expiration date associated with said public key.

25. The method according to Claim 24, wherein said communications terminal transmits a public key request to said mobile communications network if said public key has expired.

26. The method according to Claim 1, further comprising the steps of:

changing said public key at said mobile communications network; and

storing said changed public key at said communications terminal.

27. The method according to Claim 26, wherein the step of changing said public key further comprises the step of broadcasting said changed public key from said mobile communications network for a predetermined period of time.

28. A method for encrypting traffic between a generic communications network and a first communications terminal, comprising the steps of:

broadcasting a public key from said generic communications network to a plurality of communications terminals, said plurality of communications terminals including said first communications terminal;

generating a secret key at said first communications terminal;

encrypting said secret key with said public key at said first communications terminal;

transmitting said encrypted secret key from said first communications terminal;

receiving said encrypted secret key at said generic communications network;

decrypting said received encrypted secret key with a private key, said private key associated with said public key; and

encrypting said traffic with said secret key.

29. The method according to Claim 28, wherein the broadcasting step further comprises the steps of:

transferring said public key from a radio network controller to at least one base station in said generic communications network; and

transmitting said public key from said at least one base station.

30. The method according to Claim 28, wherein said broadcasting step comprises the step of transmitting said public key from a plurality of base stations in said generic communications network.

31. The method according to Claim 28, wherein said first communications terminal comprises an unidentified communications terminal.

32. The method according to Claim 28, wherein the step of broadcasting said public key further comprises the step of broadcasting information to authenticate said public key.

33. The method according to Claim 28, wherein the step of broadcasting said public key further comprises the step of transmitting, on request, information to authenticate said public key.

34. A method for encrypting communications traffic between a mobile communications network and a communications terminal, comprising the steps of:

storing two numbers associated with a Diffie-Hellman exponential key exchange algorithm and a first identifier associated with said mobile communications network at said communications terminal;

comparing said first identifier stored at said communications terminal with a second identifier received from said mobile communications network and producing a first predetermined result;

generating a first random number at said communications terminal;

generating a second random number at said mobile communications network; and

using said first and second random numbers as inputs to said Diffie-Hellman exponential key exchange algorithm, generating a third number to be used as a secret key by said communications terminal and said mobile communications network.

35. The method according to Claim 34, wherein the step of storing two numbers comprises the step of a priori pre-storing said two numbers.

36. The method according to Claim 34, further comprising the step of transmitting said two numbers from said mobile communications network upon receiving a request for said two numbers from said communications terminal.

37. The method according to Claim 36, further comprising the step of transmitting said request from said communications terminal upon said comparing step producing a second predetermined result.

38. The method according to Claim 34, wherein the step of storing said two numbers and said first identifier further comprises storing an expiration date associated with said two numbers.

39. The method according to Claim 38, wherein said communications terminal transmits a request for two new numbers associated with said Diffie-Hellman exponential key exchange algorithm if said two numbers has expired.

40. The method according to Claim 34, further comprising the steps of: changing said two numbers associated with a Diffie-Hellman exponential key exchange algorithm at said mobile communications network; and storing said changed two numbers at said communications terminal.

41. The method according to Claim 40, wherein the step of changing said two numbers further comprises the step of broadcasting said changed two numbers from said mobile communications network for a predetermined period of time.

42. A method for encrypting traffic between a generic communications network and a first communications terminal, comprising the steps of:

broadcasting two numbers associated with an exponential key exchange algorithm from said generic communications network to a plurality of communications terminals, said plurality of communications terminals including said first communications terminal;

generating a first random number at said first communications terminal;

generating a second random number at said generic communications network;

using said first and second random numbers as inputs to said exponential key exchange algorithm, generating a third number to be used as a secret key by said first communications terminal and said generic communications network;

and encrypting said traffic with said secret key.

43. A system for use in encrypting traffic between a generic communications network and a communications terminal, comprising:

an access network included in said generic communications network; and

access network means coupled to said communications terminal and associated with said access network, for storing a public encryption key associated with said generic communications network, generating a secret key, encrypting said secret key with said stored public encryption key, and transmitting said encrypted secret key to said generic communications network.

44. A system for use in encrypting traffic between a generic communications network and a communications terminal, comprising:

first network means for storing a private encryption key, distributing a public encryption key, and decrypting an encrypted secret session key;

second network means connected to said first network means, for broadcasting said distributed public encryption key, said first and second network means associated with an access network of said generic communications network; and

access network means coupled to said communications terminal and associated with said access network of said generic communications network, for receiving said broadcast public encryption key, generating a secret key, encrypting said secret key with said received public encryption key, and transmitting said encrypted secret key to said generic communications network.

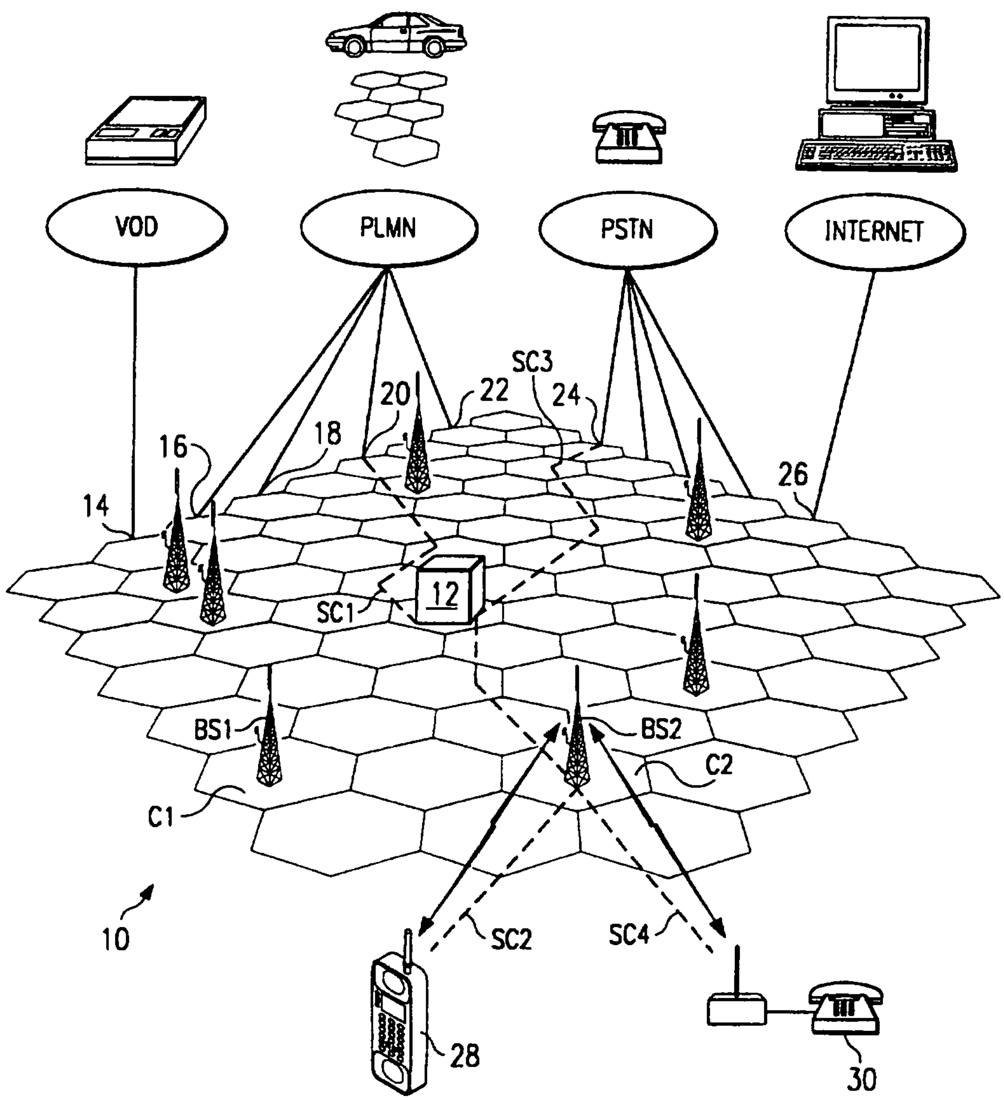

VOD
PLMN
PSTN
INTERNET
SC3
22
24
20
18
16
14
26
12
SC1
BS1
BS2
C2
C1
10
SC2
SC4
28
30

FIG. 1

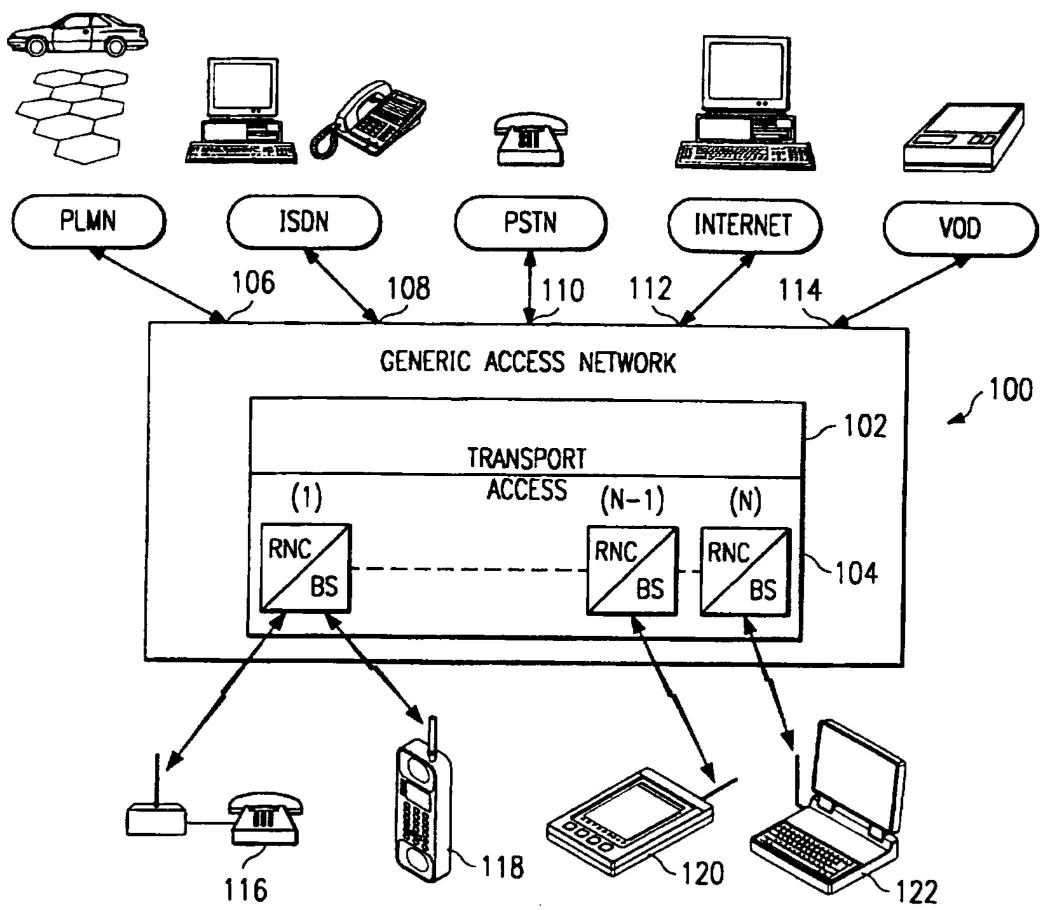

PLMN
ISDN
PSTN
INTERNET
VOD
106
108
110
112
114
GENERIC ACCESS NETWORK
100
102
TRANSPORT
ACCESS
(1)
(N-1)
(N)
RNC
BS
RNC
BS
RNC
BS
104
116
118
120
122

FIG. 2

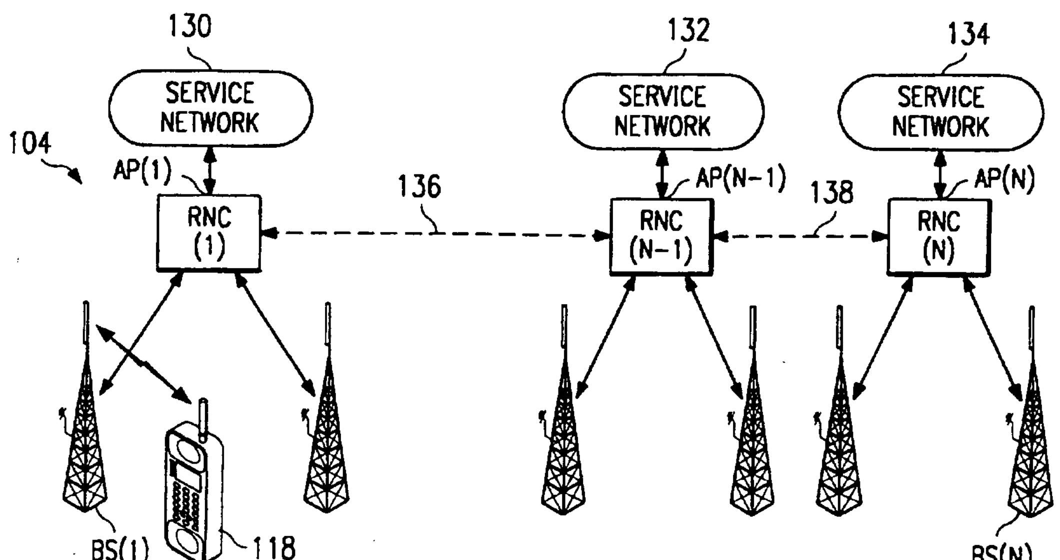

130
132
134
SERVICE NETWORK
SERVICE NETWORK
SERVICE NETWORK
104
AP(1)
136
AP(N-1)
138
AP(N)
RNC (1)
RNC (N-1)
RNC (N)
BS(1)
118
BS(N)

FIG. 3

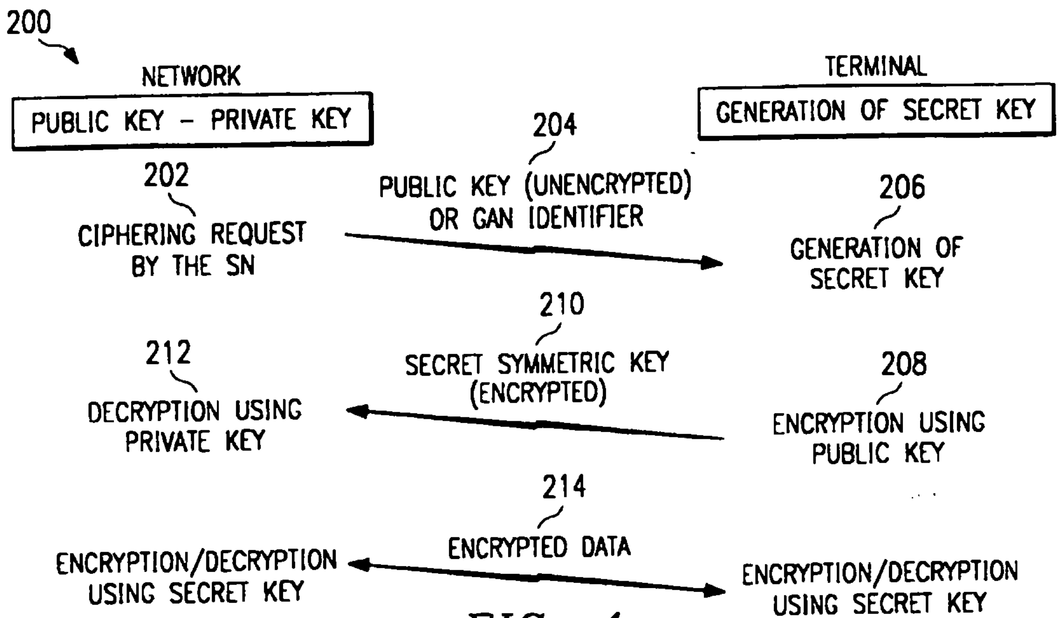

200
NETWORK
PUBLIC KEY – PRIVATE KEY
TERMINAL
GENERATION OF SECRET KEY
202
CIPHERING REQUEST BY THE SN
204
PUBLIC KEY (UNENCRYPTED) OR GAN IDENTIFIER
206
GENERATION OF SECRET KEY
210
SECRET SYMMETRIC KEY (ENCRYPTED)
212
DECRYPTION USING PRIVATE KEY
208
ENCRYPTION USING PUBLIC KEY
214
ENCRYPTED DATA
ENCRYPTION/DECRYPTION USING SECRET KEY
ENCRYPTION/DECRYPTION USING SECRET KEY

FIG. 4

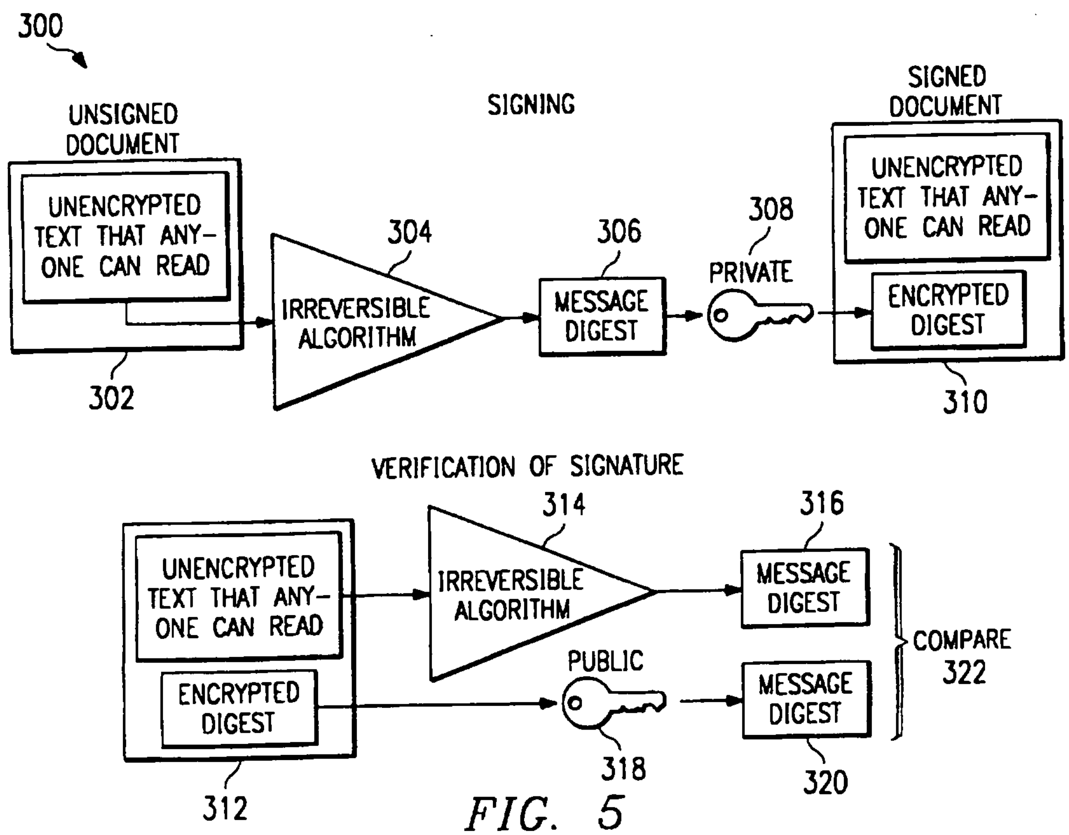

300
UNSIGNED DOCUMENT
SIGNING
SIGNED DOCUMENT
UNENCRYPTED TEXT THAT ANY-ONE CAN READ
304
IRREVERSIBLE ALGORITHM
306
MESSAGE DIGEST
308
PRIVATE
UNENCRYPTED TEXT THAT ANY-ONE CAN READ
ENCRYPTED DIGEST
302
310
VERIFICATION OF SIGNATURE
314
316
UNENCRYPTED TEXT THAT ANY-ONE CAN READ
IRREVERSIBLE ALGORITHM
MESSAGE DIGEST
COMPARE
322
PUBLIC
ENCRYPTED DIGEST
MESSAGE DIGEST
318
320
312

FIG. 5

INTERNATIONAL SEARCH REPORT

International Application No
PCT/SE 97/01407

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L9/08 H04Q7/32

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 6 H04L H04Q

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.
A	US 5 222 140 A (BELLER ET AL.) 22 June 1993 see column 4, line 57 - column 5, line 3 see column 5, line 13 - line 37	1,2,7, 10,28, 29,34, 42-44
A	GB 2 297 016 A (KOKUSAI DENSHIN DENWA) 17 July 1996 see page 19, line 11 - page 21, line 2; figure 7	28,44

-/--

[X] Further documents are listed in the continuation of box C.

[X] 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

1

Date of the actual completion of the international search	Date of mailing of the international search report
19 November 1997	02/12/1997
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 Holper, G

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/SE 97/01407

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	MEVEL F ET AL: "Distributed communication services in the Masix system" CONFERENCE PROCEEDINGS OF THE 1996 IEEE FIFTEENTH ANNUAL INTERNATIONAL PHOENIX CONFERENCE ON COMPUTERS AND COMMUNICATIONS (CAT. NO.96CH35917), CONFERENCE PROCEEDINGS OF THE 1996 IEEE FIFTEENTH ANNUAL INTERNATIONAL PHOENIX CONFERENCE ON COMPUTERS AND, ISBN 0-7803-3255-5, 1996, NEW YORK, NY, USA, IEEE, USA, pages 172-178, XP000594787 see page 174, right-hand column, line 25 - line 29 see page 176, right-hand column, line 26 - page 177, left-hand column, last line; figure 5 ---	28,43,44
A	PATENT ABSTRACTS OF JAPAN vol. 95, no. 008 & JP 07 203540 A (N T T IDOU TSUUSHINMOU KK), 4 August 1995, see abstract ---	1,34
A	EP 0 067 977 A (SIEMENS) 29 December 1982 see abstract; figure 2 -----	24

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/SE 97/01407

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5222140 A	22-06-93	NONE	
GB 2297016 A	17-07-96	JP 8195741 A	30-07-96
EP 67977 A	29-12-82	DE 3123167 C	24-02-83

Form PCT/ISA/210 (patent family annex) (July 1992)

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 1/00	A1	(11) International Publication Number: **WO 98/45768** (43) International Publication Date: 15 October 1998 (15.10.98)

(21) International Application Number: PCT/CA98/00241

(22) International Filing Date: 18 March 1998 (18.03.98)

(30) Priority Data:
08/831,696 10 April 1997 (10.04.97) US

(71) Applicant: NORTHERN TELECOM LIMITED [CA/CA]; Station A, P.O. Box 6123, Montreal, Quebec H3C 3J5 (CA).

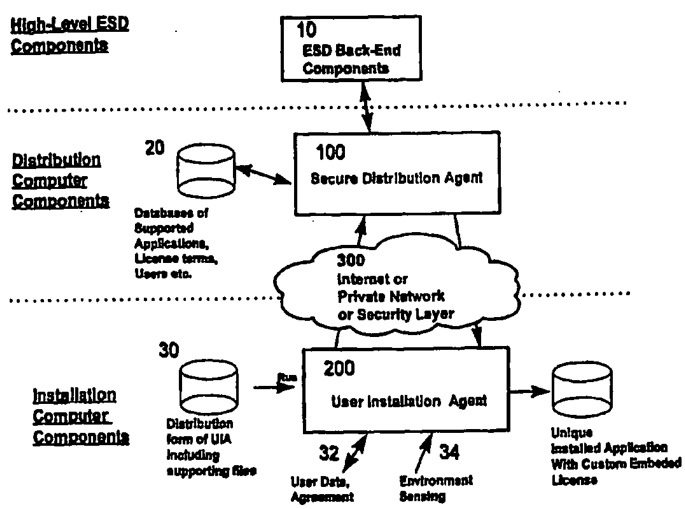
(72) Inventors: LAROSE, Gordon, Edward; 2417 Baseline Road, Ottawa, Ontario K2C 0E3 (CA). ALLAN, David, Ian; 852 Forest Street, Ottawa, Ontario K2B 5P9 (CA).

(74) Agents: MCGRAW, James et al.; Smart & Biggar, 900–55 Metcalfe Street, P.O. Box 2999, Station D, Ottawa, Ontario K1P 5Y6 (CA).

(81) Designated States: AU, CA, CN, JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: METHOD AND SYSTEM FOR NETWORKED INSTALLATION OF UNIQUELY CUSTOMIZED, AUTHENTICABLE, AND TRACEABLE SOFTWARE APPLICATIONS



(57) Abstract

A method to create, distribute and install on an installation computer a uniquely customised instance of a software application that is authenticable and traceable to a particular user. A secure distribution agent resident on a distribution computer collects indentifying information, and calculates a cryptographic signature of the software application and identifying information. The identifying information and cryptographic signature are embedded in the software application by the secure distribution agent. The software application with embedded data is transmitted via a distribution channel to the installation computer. A user installation agent resident on the installation computer manages the installation of the software application with embedded data on the installation computer. Prior to installation, the user installation agent may use the cryptographic signature to verify that the software application, and the indentifying information, are authentic and have not been tampered with.

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 SYSTEM FOR NETWORKED INSTALLATION OF UNIQUELY CUSTOMIZED, AUTHENTICABLE, AND TRACEABLE SOFTWARE APPLICATIONS

FIELD OF THE INVENTION

This invention relates to a method and system for the electronic distribution and installation to users via a network of software applications that are uniquely customized, authenticable and traceable to the individual user.

BACKGROUND OF THE INVENTION

With the increasing importance and reliance on networked computer environments such as the Internet, Electronic Software Distribution (ESD) is assuming an increased importance as a means of distributing software applications to users. The on-line infrastructures currently in place enable users to purchase and install software applications without the need for physical delivery of shrink-wrapped software. Typically, a software publisher will prepare a master of the software application for electronic distribution. A customer will then go on-line and submit an order to purchase the software application, which will be received and fulfilled by the publisher. The customer will then download the software application and install it to his/her own computer.

A disadvantage of the current on-line infrastructure is that it delivers software applications to users in a form that is identical with those found in retail stores and catalogues. Absent cryptographic protection, users can freely share the distribution form of the software amongst themselves.

Even where cryptographic protection are present, the potential for unauthorized copying is still significant because all the users possess identical copies (necessarily having identical encryption schemes) of a software application. There is in all such cases a single underlying decryption key, and in most cases this key, or an equivalent variant of it, is entered by the user, who can then share it with other users who can use it to obtain unlicensed usage of the program. There exist today bulletin boards and Internet sites devoted to the sharing of such keys, which are visited by persons interested in obtaining unpaid for usage of programs by applying such keys to copies of the applications they have obtained.

Further, even where more subtle anti-piracy schemes are in place in a software application, it is not uncommon for software "hackers" to produce "crack" programs which can be used to process a freely-distributed, limited functionality version of a program to produce a revised, fully-functional version of the same program which can be used without purchasing a license. Even the most ingenious forms of single-key mass distribution, which might involve input of one-time-only responses to a dynamic challenge to infer the key, are vulnerable to a "crack" which simply causes the application of the "true" universal decryption key. Although such "crack" involves more technical sophistication than sharing of keys as above, the distribution channels and potential effect on the product's revenues are very similar.

In addition, software applications distributed by conventional ESD techniques provide no means to police their own integrity to prevent unauthorized tampering.

Portland Software has produced an electronic software distribution system sold under the trade-mark ZipLock™ that packages software for electronic distribution over the Internet. The ZipLock™ system discloses a system that distributes, from a secure server to a client resident on the user's computer, a standard executable software application that is protected by means of a cryptographic key. Data input by the users is transmitted to the secure server and is used to construct a customized digital licence certificate that is transmitted to the user in a separate computer file. The Ziplock™ system does not provide a mechanism to detect tampering done to the executable software application itself, nor does it provide traceability if the digital licence certificate is not included with an unauthorized redistribution of the software application.

The prior art discloses a number of other systems and methods to protect unauthorized use of software electronically distributed to users. In Choudhury U.S. Pat. No. 5,509,074, there is disclosed a method of protecting electronically published materials using cryptographic protocols. A first described embodiment requires special purpose hardware to decrypt the document that is transmitted to the user. This eliminates the method from general use with personal computers used by the general public. In a second method, there is no requirement for special purpose hardware. In this method, the publisher modifies the inter-line or inter-word spaces of the document to make each document unique for each user. The unique document is then encrypted and transmitted to the user's computer. Upon receipt of the encrypted document, the user's computer will prompt the user to enter his/her secret key which is used to decrypt the document for viewing. The method disclosed by this reference does not prevent piracy, it only discourages piracy by making the pirated document traceable to the user. In addition, this reference pertains only to data files, not to the protection of executable files of any type.

In Cane U.S. Pat No. 5,416,840, there is disclosed a method and system for protecting computer program distribution in a broadcast medium, such as radio frequency public broadcast or computer network. In this reference, the method involves encrypting at least a portion of a computer program, the user being supplied with a password for use in decrypting the computer program so that the computer program can be installed and used. A unique password is generated and transmitted to the user for subsequent use in decrypting the selected software program contained on the medium. While there is disclosed a method and system for the generation, transmission and use of unique passwords that cannot be shared among different users of the software application, this reference requires the user to own proprietary hardware that eliminates it from general use with personal computer owned by the general public.

In Yuval U.S. Pat No. 5,586,186, there is disclosed a method and system for controlling unauthorized access to software distributed to users. The main components of the system are an encryptor, a user key generator, and a decryptor. The encryptor generates encryption and decryption keys, encrypts the software using the encryption keys, and stores the encrypted forms of the software of the broadcast medium, such as CD ROM. The user key generator generates a unique key using numeric representations of identifying information supplied by users and the decryption keys. The decryptor is responsible for decrypting the encrypted forms of the software using the identifying information supplied by the user, and the unique user keys. The decryption method disclosed by this reference enables a large number of different but logically similar keys to be used as decryption keys, each of which is unique to a particular user. However, this reference does not disclose a means to customize a software application with user-specific data such that the software application itself can be authenticated. Furthermore, this reference does not prevent piracy by sharing of keys; it only discourages it through traceability of keys.

SUMMARY OF THE INVENTION

The present invention pertains to a method for the electronic distribution of a software application from a distribution computer to an installation computer comprising the steps of receiving at said distribution computer identifying information, embedding said identifying information in said software application at said distribution computer to form an identifiable software application, generating a cryptographic signature for said identifiable software application, embedding said cryptographic signature in said identifiable software application to form an identifiable and authenticable software application, and transferring said identifiable and authenticable software application from said distribution computer to said installation computer.

The method and system of the present invention discloses an on-line software customization, delivery and installation scheme. Instead of distributing a software application to a user that results in the installation of a totally generic, untraceable executable file on the installation computer, the method and system disclosed herein discloses a means to create, distribute and install on an installation computer a uniquely customised instance of a software application that is authenticable and traceable to a particular user.

The method and system disclosed herein provides for a user installation agent (UIA) resident on an installation computer to establish a connection through a distribution channel to a secure distribution agent (SDA) resident on a distribution computer. The UIA and/or SDA prompt the user to input identifying information that, together with business related information such as licensing terms, etc., is used to create a unique data set that is embedded in the desired software application by the SDA. By the use of a cryptographic hash algorithm, and private/public key cryptography wherein a private key is only known to the SDA, a cryptographic signature of the desired software application and embedded data set is calculated and also embedded into the software application. The software application with embedded data and cryptographic signature is transmitted via a distribution channel to the installation computer where it is installed on the installation computer. Optionally, the installation computer may use the cryptographic signature to verify that neither the software application, nor the embedded data have been tampered with. Public key(s) used to decrypt the cryptographic signatures may be transmitted to the installation computer with the software application, or by any other means, such as e-mail, Internet bulletin boards, etc. Following installation, the embedded data and cryptographic signature are used in a variety of ways, such as to provide a means to trace the software application to the user, to police the continued integrity of the software application, to ensure that license conditions continue to be met, to perform virus checking, or automatic upgrading of the software application itself.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a system overview showing the various inputs and components of the system and method of the present invention;

Figure 2 is a data flow diagram of the structure and operation of the Secure Distribution Agent employed by the present invention;

Figure 3A is a block diagram showing details of the construction of the aggregate distribution file using a one-step cryptographic process;

Figure 3B is a block diagram showing details of the construction of an aggregate distribution file using a two-step cryptographic process;

Figure 3C is a block diagram showing details of the construction of an aggregate distribution file using a cryptographic process that is a variant of the two-step cryptographic process shown in Figure 3B;

Figure 4 is a block diagram of the structure and operation of the User Installation Agent employed by the present invention;

Figure 5 is a block diagram showing the means of extracting and authenticating embedded data from an installed distribution file;

Figure 6 is a flow chart of a first embodiment of the present invention that authenticates embedded data by means of a common encryption key;

Figure 7 is a flow chart of a second embodiment of the present invention that authenticates embedded data by means of a unique per-user encryption key; and,

Figure 8 is a block diagram showing the various uses of the installed software application delivered to the user by means of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 shows the various inputs and components of the system and method of the present invention. At the top level is shown a representation of the Electronic Software Distribution (ESD) back-end components 10, which include clearinghouses of software, software manufacturers, publishers, credit card servers, etc., all of which interact with a Secure Distribution Agent (SDA) 100 resident on a distribution computer that forms an essential part of the present invention.

The SDA 100 interfaces with these ESD back-end components 10 via the Internet or private computer network to provide payment methods support, loading of software applications from publishers, etc. The exact nature of the ESD back-end components 10 may vary without affecting the method and system of the present invention.

The SDA 100 is comprised of a system of co-operating software programs, which run in a secure environment. The nature of the secure environment is immaterial to the invention as long as it ensures the ability to protect the privacy of user data, authentication of users and possibly other third-parties, and suitable limitations on the operations which can be accessed externally. This environment might or might not be physically separated from an installation computer. The structure and operation of the SDA 100 is more fully described in Figure 2.

One of the inputs to the SDA 100 is a set of databases 20 of supported software applications, license terms, licensed users, etc. The SDA 100 transmits and receives relevant data to/from the databases 20 prior to, and during the operation of the present invention. The exact nature and content of the databases 20 is not an essential feature of the invention.

A distribution channel 300 is illustrated in Figure 1 that can comprise computer networks such as the Internet, or private network, or a security layer as required to maintain security if the SDA 100 were located in close proximity with a user installation agent (UIA) 200. Alternatively, it may contain some combination of these elements. The distribution channel 300 is used to connect the UIA 200 to the SDA 100 (and thus connect the distribution computer to the installation computer) so that information may be exchanged between these two agents, and so that an aggregate distribution file 170 (shown in Figure 2) can be distributed from the SDA 100 to the UIA 200. Though the distribution channel 300 is illustrated between the SDA 100 and the UIA 200, the system of the present invention does not require that the SDA 100 be physically distant from the UIA 200.

At a user's end is the UIA 200 which is an installation/automatic upgrade software program resident on the installation computer. This program is used to communicate via the distribution channel 300 to the SDA 100, and to perform the required operations, more fully described below, on the installation computer. Though normally one UIA 200 would be required for each supported software application, persons skilled in the art would be familiar with the capability to develop a UIA 200 that would support multiple software applications. Also shown in Figure 1 is a distribution form 30 of the UIA 200, including support files. The nature of the distribution form 30 of the UIA 200 is immaterial to the operation of the present invention. Any of CD ROM, World Wide Web (WWW) download, floppy diskette, etc. could be used.

The UIA 200 accepts data 32 input from the user, such as name, address, payment options, etc., as well as data pertaining to the acceptance of an end-user license. Environment sensing data 34 such as speed of CPU, size of hard disk, speed of modem, etc. may also be input to the UIA 200 for processing. The identifying information processed by the UIA 200 may include any information pertaining to the purchaser, the seller, the installation agent, date, serial number, license specifics, etc. This data may be used for the automatic registration of the desired software application with a publisher or its commercial proxies.

As noted above, the identifying data 32, 34 constitutes identifying information concerning the user, its computer, etc. The identifying data 32, 34 is processed by the UIA 200 and transmitted to the SDA 100 via the distribution channel 300. Of course, it is understood that the identifying information does not necessarily have to be transmitted to the SDA 100 by means of the distribution channel 300. For example, the identifying information may be locally entered into the SDA 100 by an agent using information received verbally, in writing, or in some other non-electronic manner. The SDA 100 combines the identifying data 32, 34 with the data stored in the databases 20 to produce an aggregated distribution file 170 that is uniquely customized, authenticable, and traceable to the user. The aggregate distribution file 170 is transmitted via the distribution channel 300 to the UIA 200. The output from the UIA 200 is a uniquely customized software application 15 (to be referred to below as an "installed aggregate distribution file") installed on the installation computer, with identifying information embedded therein.

Though the description of the present invention implies that the "user" is an individual user of the software application 15 to be installed on a personal computer, persons skilled in the art will appreciate that the present invention would also operate in the context of a networked end-user environment, where the "user" was a network administrator responsible for installing software on a central server for use by a number of end users.

Figure 2 is a data flow diagram of the structure and operation of the SDA 100 employed by the present invention. An original distribution file 130 is shown as an input to a conversion program 110. In the envisioned implementation, the original distribution file 130 is input to the SDA 100 by the databases 20 shown in Figure 1. It is understood that the original distribution file 130 does not necessarily have to be input to the SDA 100 by the databases 20, since the original distribution file 130 may already be resident on the distribution computer containing the SDA 100. The conversion program 110 has, as additional inputs, the data 140 to be embedded in the distribution file 130, and required public/private cryptographic key pairs 150. The embedded data 140 is produced by a user interaction program 120 which interacts with the user through the UIA 200 to receive identifying data 32, 34 (shown in Figure 1) as well as data from the databases 20 of supported software applications, license terms, licensed users, etc.

While the embedded data 140 can be of any form and content, it is anticipated that the embedded data 140 will contain information enabling the software application 15 to be traceable to an individual user and license transaction. For example, the embedded data 140 can include a unique serial number used to identify the aggregate distribution file 170 to be distributed to the user. The would eliminate serial number fraud that is common in the software industry, whereby current software applications can only perform simple validity checks, which can be fooled by widespread fraudulent re-use of a single valid serial number. The embedded data 140 may take the form of a complete license agreement customized to the individual user, including user name, address, software serial no., license terms, etc. Records of the user information collected by the user interaction program 120 may be kept by the databases 20.

The output of the conversion program 110 is an aggregate distribution file 170 which contains both the contents of the original distribution file 130, the embedded data 140, as well as a cryptographic signature of the embedded data 140 and the original distribution file 130. The aggregate distribution file 170 is then transmitted via the distribution channel 300 to the UIA 200. The UIA 200 then installs the aggregate distribution file 170 on the installation computer. Once the aggregate distribution file 170 is installed, it takes the form of an installed aggregate distribution file 15.

By means of its connection with the UIA 200, the SDA 100 can negotiate arbitrary license terms with the user, display an End-User License Agreement (EULA), confirm acceptance of that agreement, and automatically perform on-line registration of the software based on the already-established identity of the user and the specific license terms. Subject to commercial and legal considerations, an SDA 100 could offer different pricing and license terms, and possibly different executable versions, to users in different countries, for example. In addition, differential pricing based on attributes of the installation computer such as CPU power could be provided.

The SDA 100 does not require intelligence within itself for functions such as establishing that a user's stated address and credit-card number are valid, consistent, and within a given geographical area. Such functions may be undertaken by the high-level ESD components 10 illustrated in Figure 1.

Figure 3A shows the procedure for constructing the aggregate distribution file 170 in greater detail. For the sake of illustration, the original distribution file 130 is assumed to have a structure including header information and different types of internal sections for code, static data and so on, such as a Windows™ 'Portable Executable' (PE) program file. One of ordinary skill in the art can appreciate that the method and system of the present invention can be applied to a number of different file formats. Similarly, the inputs 140, 151 to the conversion program 110, and output 170 from the conversion program 110 are illustrated to be computer files, but they could be in-memory images, streams from other processors, etc.

A typical sequence of steps run by the SDA 100 to construct the aggregate distribution file 170 is described below.

1. The conversion program 110 is run, as a result of the user interaction program 120 having determined that a conversion is required i.e. that a delivery of a particular aggregate distribution file 170 according to the method of the present invention is authorized, and that the required embedded data block 140 has been constructed. All subsequent steps are executed by the conversion program 110 unless otherwise indicated.

The object is to obtain what is often referred to as a "digital signature", or "cryptographic signature" which inherently has two aspects:

(i) By the use of a cryptographic hash algorithm, the production of a cryptographic fingerprint that uniquely corresponds to the data "ed" 130, 140; and

(ii) Protection of that cryptographic fingerprint by encrypting it with a private key, such that the recipient of the cryptographic fingerprint may, by using a public key and the cryptographic algorithm, verify that the data "ed" 130, 140 is intact, without having the ability to generate a new cryptographic fingerprint, and plausibly change the data.

These two steps are essential to realize the advantage of the present invention, since without both steps a third party may intervene and alter data without the recipient being able to detect it. This procedure is to be distinguished from simply encrypting the data "ed" 130, 140, which is a step that is not necessary to the operation of the present invention since there is no way to plausibly alter the data 130, 140 without such alterations being detectable.

2. The input/output logic 111 of the conversion program 110 reads in the desired original distribution file 130, its corresponding cryptographic private key 151, and the data to be embedded 140. Though not required by the conversion program 110, a public key 152 may be passed through in order that it may be added to the aggregate distribution file 130. Utilizing cryptographic hash algorithms 112 and Public-Private key (PPK) encryption algorithms 113, a cryptographic signature 174 is produced. The basic steps of this process are:

2.1 Apply a one-way hash function "hf" to the data "ed" 130, 140 producing a cryptographic fingerprint "edh", that is edh = hf(ed). The requirements for this cryptographic fingerprint are as follows: (i) that it produce a reasonably compact result i.e. size(edh) << size(ed), and preferably a fixed-length result; (ii) that the fingerprint alone cannot be used to ascertain the original data block back i.e. there is no back-hash function "bhf" such that bhf(edh) = ed; (iii) that it be extremely sensitive to changes in "ed"; say, that a single-bit change in "ed" changes on average 50% of the bits in "edh", and (iv) that it is extremely difficult to construct a false embedded data block "fed" which produces the same fingerprint as "ed", that is hf(ed) = hf(fed). There are a number of algorithms which satisfy these requirements, such as MD5 (Message Digest 5) and SHA (Secure Hash Algorithm). Other algorithms that also meet the above criteria that may be employed by the present invention.

2.2 Encrypt the cryptographic fingerprint "edh" using the private key 151 "prk" and a public/private encryption function "ppef" to produce a cryptographic signature "edf" 174, that is: edf = ppef(prk, edh). The requirements for the encryption function "ppef" are as follows: (i) that it produce a result not substantially larger than its input; (ii) that it effectively protect relatively short data sets, since "edh" will be bytes long rather than kilobytes long; (iii) that is computationally infeasible to use the public key 151 ("puk") and the cryptographic signature "edf" 174, or multiple instances of "edf" 174 (which will be visible on the installation computer) to infer the private key "prk", that is, there is no cracking function "cf" such that: puk = cf(edf,puk); (iv) that there is no conceivable means of replicating the behaviour of "ppef" using "prk" without in fact possessing both "ppef" and "prk". In principle, "ppef" can be inferred from its corresponding decryption function, so "prk" is the important secret in practical terms; (v) that the corresponding public-key decryption function "ppdf" have acceptable performance on a typical installation computer for the pertinent file sizes. Note that if a specific ppef/ppdf is chosen for security reasons and does not yield acceptable performance, the encryption could be applied to only a portion of the selected files and still offer the same benefits; (vi) that it be suitable (preferably, via established cryptanalysis) for specific application in this domain i.e. digital signatures. There are a number of algorithms which might satisfy these requirements, such RSA and those of Rabin and ElGamal. The careful selection of implementation parameters can help attain required security and performance.

3. The cryptographic signature 174 from step 2.1, and the data to be embedded 140 are inserted into the original distribution file 130 to produce the aggregate distribution file 170. This insertion is not a simple copying of bits into the middle of a file, since it must be compliant to the format requirements of the particular file types. For example, headers may have to be updated to identify the new data etc.

The system and method of the present invention does not require that the embedded data 171 or the cryptographic signature 174 be positioned in any particular manner in the aggregate distribution file 170. What is necessary is that: (i) the software on the installation computer, and the UIA 200 in particular, be able to locate the embedded data 171 and cryptographic signature 174, and (ii) that the aggregate distribution file 170, after it is installed on the installation computer, be able to perform its intended function; for example, if it is an executable file, that it still conform to structural and other platform requirements so it can load and run on the installation computer as it might have before the conversion process. For example, if the file were in a format common to current computers containing an Intel™ microprocessor and running a Microsoft™ Windows™ operating system, the conversion program 110 could inspect the "header" section of the original distribution file 130 to determine where there were sections containing static data so as to avoid sections containing executable code. A static data section would be selected and a suitable location for the embedded data block 171 and cryptographic signature 174 would be found or created. This would be done by, for example, (i) determining that an existing static data block had unused capacity sufficient to add the data, (ii) allocating a new static data block, or (iii) expanding an existing static data block.

The method illustrated in Figure 3A discloses a one-step process wherein cryptographic signature 174 is ascertained for the original distribution file 130 and the embedded data 140. An optional method, such as that illustrated in Figure 3B, would be to employ a two-step process wherein a cryptographic signature 172 of the embedded data 171 is first produced using the same algorithm described in step 2 above. This embedded data cryptographic signature 172 is then itself embedded into the original distribution file 130. The original distribution file 130, embedded data 171, and embedded data cryptographic signature 172 are then input to the second cryptographic step, wherein an overall cryptographic signature 176 is ascertained using the same algorithm described in step 2 above. The benefit of the two step process is that it augments the capabilities of the system and method of the present invention to authenticate and detect tampering in the software application installed on the installation computer. For example, separate cryptographic public/private key pairs could be provided for the two cryptographic signatures 172, 176. Furthermore, the two-step process allows the embedded data 171 to be extracted and authenticated, even if the original file contents 173a, 173b have been corrupted.

Another alternative is to construct the aggregate distribution file 170 using a variation of the two-step cryptographic process wherein a first cryptographic signature 175 is made of only the original file contents 173a, 173b, and a second cryptographic signature 172 is made of the embedded data 171. This is illustrated in Figure 3C. This scheme has all the advantages of the two-step process illustrated in Figure 3B, and also allows for separate authentication of the embedded data 171 and the original file contents 173a, 173b. This would allow a user to verify that original distribution file 130 provided by the publisher had not been altered by the on-line installation process disclosed by the present invention.

One of ordinary skill in the art will appreciate that any of the cryptographic signatures 172, 174, 175, 176 shown in Figures 3A, 3B and 3C do not have to be produced using the same set of cryptographic public/private key pairs, or even the same cryptographic algorithms. As well, it is not necessary that the cryptographic signatures 172, 174, 175, 176 be calculated each time an aggregate distribution file 170 is distributed to a user. The SDA 100 could maintain a database of partially-precomputed signatures to speed up the related calculations. The availability of cryptographic hardware support such as RSA co-processors in the installation computer, could be used to attain good responsiveness with maximal security. As well, it is not essential that the aggregate distribution file 170 be constructed in its entirety by the SDA 100. What is necessary is that the aggregate distribution file 170 be derivable in its entirety by the UIA 200.

Figure 4 illustrates the structure and operation of the UIA 200 which consists of a transient installation index 204, a transient installation input fileset 205, and a UIA proper executable software program 203. One skilled in the art will appreciate that there are many ways in which the UIA program 203 could be implemented. Since a significant part of the UIA's 200 functionality involves user interaction and dialog with the SDA 100, options for the implementation of the UIA 200 include either making it an adjunct to a World Wide Web (WWW) browser, or implementing it as a stand-alone program which itself embeds or invokes already-present browser capability on the installation computer.

A typical execution sequence of the UIA 200 is described below:

1. After the UIA program 203 and its support data 204, 205 have been copied onto the installation computer, the user runs the UIA program 203. Note that the UIA program 203 could also have been initiated remotely e.g. sent as an active program within a browser framework by a WWW server. Unless otherwise stated, all subsequent steps are executed by the UIA program 203.

2. The installation index 204 and installation input fileset 205 are read by the installation computer to determine the particular default SDA 100 appropriate for the installation

of the desired software application (known as the "installed aggregate distribution file") 15.

3. The installation computer is examined to determine the probable means of establishing communications with the SDA 100, for example, the presence of TCP/IP network interfaces, modems etc. If no such means are found, the program optionally assists the user to find parameters which will work properly, then fails with a warning. This is because access to the SDA 100 is essential for operation of the invention.

4. The user 1 is prompted with the default data from steps (2) and (3) above, i.e. informed where the UIA program 203 will look for the desired SDA 100, and over what sort of distribution channel 300. The user 1 is then given an opportunity to change this information, either for commercial reasons (e.g. maybe an SDA has changed names or locations), or for technical reasons (e.g. the user does not have working TCP/IP connectivity and wants to use a straight modem link, perhaps via an 800 number.)

5. Via the distribution channel 300, the UIA program 203 establishes contact with SDA 100. If this cannot be done, the UIA program 203, after optionally helping the user determine parameters which will work properly, fails with a warning. While the security of the distribution channel 300 is optional to the operation of this invention, it is expected that the distribution channel 300 will support appropriate protocols to protect the SDA 100 from fraud. A common protocol supporting authentication and privacy, such as Secure Sockets Layer (SSL) is appropriate.

6. The UIA program 203 acts as an intermediary between the user and the SDA 100, enabling the user 1 to establish any legitimate agreement which the SDA 100 supports with respect to the desired installed aggregate distribution file 15. The UIA program 203 also has the ability to determine whether the available system resources of the installation computer meet the requirements of the desired installed aggregate distribution file 15.

There are no technical limits to the variety of options that can be displayed to the user, the questions the user might be asked, the data that might be gathered about the installation computer, etc. Since the SDA 100 is being run throughout the data gathering, data embedding, software distribution and software installation process, the system and method of the present invention can employ various levels of cryptography without the user ever being informed of the cryptographic keys, or any information from which they could be derived. This is unlike other electronic delivery systems which typically require subsequent off-line entry of 'secret keys' or derivates thereof which have been explicitly divulged to the user. Of course, public keys used for the authentication of cryptographic signatures are an exception in that the user may be able to determine them easily, however this is not a security issue since they have no fraudulent application.

7. Assuming that the user 1 meets all the criteria set out by the SDA 100, the SDA 100 will determine a specific set of files that must be transmitted to the UIA 200 to complete installation on the installation computer, notably including at least one aggregate distribution file 170 (shown in Figures 3A-3C). It is immaterial to the system and method of the present invention what is the nature of the agreement entered into between the user 1 and SDA 100, or how it is validated. That is the responsibility of the SDA 100 and its subtending commercial systems 10, if any. Most importantly, the UIA 200 does not and cannot itself decide whether an agreement has been reached between the user and the SDA 100. The UIA 200 does not have, and should not have, access to all the information required to complete the installation, except through interaction with the SDA 100.

8. The SDA 100 transmits an index of the required distribution files to the UIA 200 via the distribution channel 300. The UIA 200 uses this index to augment its own local index 204 forming a complete index for the upcoming installation.

9. The SDA 100 constructs one or more aggregate distribution files 170 and any other files required for the installation, and transmits these files to the UIA 200 via the distribution channel 300.

10. Using its local index and support files 204, 205 the UIA program 203 completes the installation of the installed aggregate distribution file 15 in a manner compliant with the platform of the installation computer. In particular, the UIA 200 installs the aggregate distribution file 170 such that the cryptographic signature 174 and the embedded data 171 are unaffected. Once the aggregate distribution file 170 is installed on the installation computer, it is referred to as an installed aggregate distribution file 15. The UIA program 203 will also perform other system updates 212 as necessary, such as updating the operating system registry (in the case of Windows 95™), and installing any additional application files. Other optional operations, such as leaving an appropriate 'uninstall' utility, may also be involved.

12. If an error should occur, the UIA program 203 may signal the SDA 100 to re-initiate the installation. If no error has occurred, the UIA program 203 signals the SDA 100 that all required data has been received. This could, for example, be used as the trigger signal for the SDA 100 to commit to a financial transaction. Leaving the financial commit to this late part of the process minimizes the probability of the user being charged for a software application which has not been successfully installed, thus reducing one cause of customer frustration.

13. The UIA program 203 deletes any transient files, indices etc. that it might have placed on the installation computer.

14. The UIA program 203 disconnects from the SDA 100 and the distribution channel 300 and exits.

Upon successful completion of the optional authentication procedures described in further detail below, the user can then run the installed aggregate distribution file 15 on the installation computer. It should be understood that the authentication procedures described below can be done either before or after the installation is completed.

The method and system of the present invention would diminish disputes arising from software which is purchased but not successfully installed. The UIA 200 can detect and warn the user if the installation computer had inadequate resources to run the desired software application, before any financial transaction has been made. Further, the final financial commitment to purchase the software application by the user can be done late in the installation process so that the probability of the financial transaction being successful, but the installation itself failing, would be low.

One of ordinary skill in the art will appreciate that the UIA 200 may be distributed to users in a mass-produced media form containing the original distribution file 130, or a derivative thereof not subject to successful fraudulent re-use through simple copying. In this scenario, the SDA 100 would transmit to the UIA 200 only the incremental information which the UIA 200 would require to construct the aggregate distribution file 170 and complete the installation. Any attempts to pirate the software application can be defeated by ensuring that the distribution form of the UIA 200 contains an incomplete set of executable files, thereby requiring essential data from the SDA 100 to be capable of executing on the installation computer.

Figure 5 illustrates the means of authenticating and extracting user data from an installed aggregate distribution file 15 to verify that neither the original file contents 173a, 173b, nor the embedded data 171 have been tampered with. This step is optional to the operation of the present invention because the installed aggregate distribution file 15 may be run by the user without authentication. It should be understood that the authentication procedures described below can be done either before or after the installation is completed. If authentication is done prior to installation on the installation computer, then the following procedures are directed by the UIA 203 to the aggregate distribution file 170, instead of the installed aggregate distribution file 15.

The process illustrated in Figure 5 is in relation to an installed aggregate distribution file 15 constructed using the two-step process illustrated in Figure 3B. The principles of authenticating and extracting user data from an installed aggregate distribution file 15 constructed using the one-step cryptographic process illustrated in Figure 3A, or the variant of the two-step process illustrated in Figure 3C, are the same as those described below, with appropriate modifications, depending on the nature of the cryptographic signatures to be compared.

Though a separate authentication and reading program 400 is shown performing the functions of authentication and reading of embedded data 171, a person skilled in the art will appreciate that these functions need not be embodied in such a stand-alone program, and could be incorporated as functions of other programs, such as the UIA 200, a license-checker, a virus-checker, a program loader, a copy program, etc. A typical execution sequence of the authentication and reading program 400 is described below:

1. The authentication and reading program 400 is run, either by a user or by automatic invocation from another program such as the UIA 200. Unless otherwise indicated, the following steps are all executed by authentication and reading program 400.

2. Determine which installed aggregate distribution file 15 to process, either by prompting the user or having this passed as a parameter by the UIA 200. Also determine (if derivable therefrom in the particular implementation, as opposed to being contained in the file itself), which particular public key 152 is applicable to this installed aggregate distribution file 15.

3. Open the installed aggregate distribution file 15 in question and check that it meets the applicable format requirements. For example, a given implementation might support executable (EXE) and dynamic link library (DLL) files

in the 'PE' format for Intel™ processors. If the installed aggregate distribution file 15 fails these basic checks, or is not found, the authentication and reading program 400 fails with an appropriate warning.

4. Examine the file to determine the location of the overall cryptographic signature 176, the embedded data cryptographic signature 172, and the embedded data 171. The installed aggregate distribution file 15 can be formatted in various ways to support this, such as including pointers to these sections in the file header. If applicable in the particular implementation, (i.e. the public key 152 is included in the file as opposed to being otherwise determined the authentication and reading program 400), find and extract the required public key 152.

If any of the above steps fail, the authentication and reading program 400 fails with an appropriate warning.

5. Use the public key 152 to decrypt the overall cryptographic signature 176 into its unencrypted form 176a (the decrypted remote overall fingerprint).

6. Using the same known cryptographic signature algorithm as was employed by the SDA 100, calculate a local version 176b (the locally calculated overall fingerprint) of the overall cryptographic signature. This calculation will necessarily exclude the overall cryptographic signature 176 itself i.e. cover all parts of the installed aggregate distribution file 15 except 176, in order that the locally calculated overall fingerprint 176b will not depend on itself.

7. Compare the locally calculated overall fingerprint 176b to the decrypted remote overall fingerprint 176a. If they differ, the authentication and reading program 400 will fail with a warning that the installed aggregate distribution file 15 has been corrupted. At this point, the UIA 200 may be invoked to contact the SDA 100 to re-acquire the installed aggregate distribution file 15.

8. Extract the embedded data 171 and present it graphically to the user, if the program has been user-invoked,

or pass it in message form to the invoker routine, if software-invoked.

9. Use the public key 152 to decrypt the embedded data cryptographic signature 172 into its unencrypted form 172a (the decrypted remote embedded data fingerprint).

10. Calculate a local version 172b (the locally calculated embedded data fingerprint) of the embedded data cryptographic signature 172 using the same known cryptographic signature algorithm as the SDA 100 used.

11. Compare the locally calculated embedded data fingerprint 172b to the decrypted remote embedded data fingerprint 172a. If they differ, the authentication and reading program 400 will fail with a warning that the embedded data 171 has been corrupted.

A similar procedure of comparison would be followed in respect of the cryptographic signature 174 if the one step process illustrated in Figure 3A had been followed. As well, a similar procedure of comparison would be followed in respect of the original file contents cryptographic signature 175 if the variant of the two-step cryptographic process illustrated in Figure 3C had been undertaken.

Figure 6 is a flow-chart of a summary of the procedures described in relation to Figures 2, 3A, 3B, 3C, 4 and 5. It should be noted that the public key 152 used to authenticate the integrity of the installed aggregate distribution file 15 could be delivered to the UIA 200 by any means since it is not a secret and might be useful for more than one purpose. For example, the public key may be embedded in the aggregate distribution file 170, it may be explicitly sent to the user as a separate file or message, or it may be obtained automatically by the installation computer from a network trusted authority (e.g. Verisign™ Inc.)

Figure 7 is a flow-chart of another set of procedures that may be employed in accordance with the present invention, whereby the original file contents 173a, 173b are encrypted using a unique private key calculated by the SDA 100 for this particular transaction. A record of this unique private key is kept by the SDA 100, and the corresponding unique public key is transmitted with the aggregate distribution file 170 via the distribution channel 300 to the UIA 200. The UIA 200 will decrypt the aggregate distribution file 170 using the public key. For security reasons, it is preferred that this public key not be permanently stored on the installation computer. Instead, the unique public key would exist only in the computer's Random Access Memory (RAM) for the duration of the installation. This makes usable redistribution of the aggregate distribution file 170 practically impossible.

Although the present invention has been described with reference to the preferred embodiments, one of ordinary skill in the art will recognize that a number of variations, alterations and modifications are possible. In Figure 8 there is an illustration showing the various uses of the installed aggregate distribution file 15. After installation and authentication by the UIA 200, the installed aggregate distribution file 15 may run normally without making use of the embedded data 171 in any way. To ensure licence compliance, the installed aggregate distribution file 15 may also be run in association with a license-enforcement program that verifies that any license terms comprising part of the embedded data 171 are being complied with. The embedded data 171 and cryptographic signatures 172, 174, 175, 176 (depending on the manner in which the aggregate distribution file 170 was constructed) may also be used as an input to a virus checker that may perform an integrity check on the installed aggregate distribution file 15 by using the public key 152 and the same known cryptographic signature algorithm as was employed by the SDA 100. Each time the installed aggregate distribution file 15 is run, the authentication and reading program 400 shown in Figure 5 may also be run, either by itself, or in association with an authenticating loader that would reject tampered files, and would not permit a tampered installed aggregate distribution file 15 to be run. The embedded data 171 may also be used simply for display to the user.

The method and system disclosed herein can also be used to upgrade an installed aggregate distribution file 15 present on an installation computer. In this case, the UIA 200 and the SDA 100 would verify of the license status of the installed aggregate distribution file 15 present on the installation computer, and then invoke the method and system disclosed herein to construct, deliver and install an upgraded version of the installed aggregate distribution file 15 to the installation computer. The capability to invoke the upgrading feature of the present invention could be done at the request of the user, or it could be invoked automatically upon detection by the UIA 200 of the availability of a new version of the original distribution file 130.

The uniqueness of the installed aggregate distribution file 15 can be used to restrict its operation to a specific central processing unit (CPU) on the installation computer. The identification of the CPU for these purposes would be done by the UIA 200 during the stage of gathering data 32, 34 for transmission to the SDA 100.

The SDA 100 and UIA 200 disclosed herein are not restricted to being invoked at the time of installation or upgrading of the installed distribution file 170. For example, in a computer game environment, the SDA 100 and UIA 200 could be invoked when the user reaches a certain point in the game, giving the user the option to purchase additional functionalities or levels for the game.

This disclosure does not presuppose that the UIA 200 does not possess added intelligence to increase the functionality of the present invention. For example, the UIA 200 may possess the intelligence to find and recognize separate Personal Digital Certificates on the installation computer which establish his identity for purposes sufficient to authorize all, or part of, the transaction in question. Such Personal Digital Certificates and their method of application would conform to established standards such as those used by commercial certificate provider Verisign™ Inc. In addition, the UIA 200 could possess the intelligence to find and recognize digital "coupon" certificates which establish that the user has some specific privilege, such as an entitlement to a specific price for a piece of software, or one which establishes his membership in a specific group, such as a company. In addition, the UIA 200 could locate pre-existing files installed according to the method of the present invention, and examine the embedded data 171. If the UIA 200 determines that there is license information present which may affect the terms of the transaction, or which may indicate a user's likely interest in, for example, an upgrade, the UIA 200 can transmit this information to the SDA 100 so that it can suitably mediate the transaction, advertise an upgrade, etc. A typical example of this would be examining a word-processing application installed in accordance with the present invention to determine that the user is entitled to a free upgrade, which the present invention can then proceed to install.

In another set of variations of the invention, the installed aggregate distribution file 15 is one which uses the principles of Nortel Algorithmic Authorization (NAA), as disclosed in U.S. Patent Application No. 08/674,037 to add robust self-policing of its own integrity. In a first variation, the run-time NAA algorithms, which already have the capability of using the installed aggregate distribution file's 15 own code as an input required for proper operation, and thus of forcing catastrophic failure in the event of tampering, have the scope of this input expanded to include an in-memory copy of one or more of the data items added by the SDA 100, such as the overall cryptographic signature 176.

In a second variation, the "launch stub" component could go further, extracting and decoding the embedded data 171 in the installed aggregate distribution file 15, and comparing the license terms therein (e.g. a specific CPU identified by, say, a certain physical Media Access Control address on a network card) to those it found by examining its current environment. In accordance with the principles of Nortel Algorithmic Authorization, the "launch stub" would not have to "decide" whether to proceed, since such decision-points are obvious attack points for 'hackers' wishing to defeat security mechanisms. Rather, it could modify data upon which proper program operation depended, in such a way that the program would continue to run properly only if said data corresponded to the proper environment per the license. As for the first variation, the application would have been pre-constructed for the specific instance, as per the patent-pending technology, in such a way that its proper flow of control used input data that was initially 'incorrect' in just such a way as to be 'corrected' only by application of the correct license data, or a simple derivative thereof.

The invention disclosed herein does not necessarily alter the functionality of the installed form of the installed aggregate distribution file 15, it only adds information and authenticability to it. However, there are a number of means by which the behaviour of the installed aggregate distribution file 15 can be mediated in new ways enabled by this invention. In one variation, the SDA 100 would have access to a variety of executable forms for a given program, or to software routines which would dynamically construct variant forms, in order to produce a program which meets particular customer function/cost requirements, and/or which actively binds itself to very specific license terms. For example, in the Microsoft Windows™ environment, different behaviour could be embodied by different Dynamic Link Libraries (DLLS) which could be selectively included.

In another variation, the initial executable form of the program file would have specific functional and license-binding choices built-in, and the SDA 100 would inject (possibly authenticable) data into the executable file which caused it to exhibit the desired behaviour. In yet another variation, the SDA 100 could make use of routines with detailed knowledge of specific program structures in order to add variant code to a pre-existing executable program which was not explicitly designed to accommodate such variation.

The described embodiments of the present invention focus on a single "core" file of a specific file type as the cornerstone of a software application's installation and security. However the method of the present invention may certainly be applied to more than one file or file type in a particular case. For example, all of the static files associated with an installed software application could receive embedded information such that they were all authenticable and associable with the particular application and installation instance.

We Claim:

1. A method for the electronic distribution of a software application from a distribution computer to an installation computer comprising the steps of:

a. receiving at said distribution computer identifying information;

b. embedding said identifying information in said software application at said distribution computer to form an identifiable software application;

c. generating a cryptographic signature for said identifiable software application;

d. embedding said cryptographic signature in said identifiable software application to form an identifiable and authenticable software application; and

e. transferring said identifiable and authenticable software application from said distribution computer to said installation computer.

2. The method of claim 1, wherein the step of generating a cryptographic signature for said identifiable software application includes the steps of

a. applying a one-way hash function "hf" to the identifiable software application "ed" producing a hash result "edh", where edh = hf(ed); and

b. encrypting the hash result "edh" using a cryptographic key to obtain a cryptographic signature.

3. The method of claim 2, wherein the one-way hash function is generated using any one of a Message Digest 5 (MD5) algorithm and a Secure Hash Algorithm (SHA) algorithm.

4. The method of claim 2 or 3, wherein the step of encrypting the hash result "edh" includes the step of using a public/private encryption function "ppef" and a private encryption key "prk" to encrypt the hash result "edh" to produce a cryptographic signature "edf" where edf = ppef(prk, edh).

5. The method of claim 4, wherein the public/private encryption function "ppef" is generated using any one of an RSA algorithm, a Rabin algorithm and an ElGamal algorithm.

6. The method of claim 1, 2, 3, 4 or 5, wherein the distribution computer and the installation computer are connected by the Internet.

7. The method of claim 1, 2, 3, 4, 5, or 6, wherein the identifying information received at said distribution computer is transmitted from said installation computer.

8. A method of receiving electronically at an installation computer a software application distributed from a distribution computer comprising the steps of:

a. receiving an identifiable and authenticable software application from the distribution computer, the identifiable and authenticable software application having embedded therein the identifying information and a cryptographic signature of the identifiable and authenticable software application; and

b. installing the identifiable and authenticable software application at the installation computer.

9. The method of claim 8, wherein prior to the step of receiving an identifiable and authenticable software application from the distribution computer, the installation computer transmits identifying information to the distribution computer.

10. The method of claim 8 or 9, wherein prior to the step of installing the identifiable and authenticable software application, the installation computer authenticates the integrity of the software application.

11. The method of claim 10, wherein the installation computer uses the cryptographic signature to authenticate the integrity of the software application.

12. A method for the electronic distribution of a software application from a distribution computer to an installation computer comprising the steps of:

a. receiving identifying information at said distribution computer;

b. embedding said identifying information in said software application at said distribution computer to form an identifiable software application;

c. generating a cryptographic signature for said identifiable software application;

d. embedding said cryptographic signature in said identifiable software application to form an identifiable and authenticable software application;

e. transferring said identifiable and authenticable software application from said distribution computer to said installation computer; and

f. installing said identifiable and authenticable software application at said installation computer.

13. The method of claim 12, wherein the distribution computer and the installation computer are connected by the Internet.

14. The method of claim 12 or 13, wherein the identifying information received at said distribution computer is transmitted from said installation computer.

15. A software distribution computer for distributing an identifiable and authenticable software application to a user comprising:

a. a communications link between said software distribution computer and said user;

b. a storage device for storing a software application for distribution;

c. a communications interface in communication with said link, for receiving identification data from said user, and for transferring said identifiable and authenticable software application to said user;
d. means for embedding identification data received from said installation computer in said software application to form an identifiable software application;

e. means for generating a cryptographic signature for said identifiable software application; and

f. means for embedding said cryptographic signature in said identifiable software application to form said identifiable and authenticable software application.

16. A software installation computer for receiving an identifiable and authenticable software application distributed by a distribution computer:

a. a communications link between said software installation computer and said software distribution computer;

b. a storage device for storing identification data, and for storing an installed software application;

c. a computer communications interface in communication with said link, for transferring said identification data, and for receiving said identifiable and authenticable software application, the identifiable and authenticable software application having embedded therein the identification data, and a cryptographic signature of the identifiable and authenticable software application;

d. means for installing said identifiable and authenticable software application on said computer storage device.

17. A software distribution computer for distributing an identifiable and authenticable software application from a distribution computer to an installation computer comprising:

a distribution computer;

an installation computer;

a communications link between said installation computer and distribution computer;

said distribution computer comprising:

a. distribution computer storage device for storing a software application for distribution;

b. a distribution computer communications interface in communication with said link, for transferring an identifiable and authenticable software application to said installation computer and for receiving identification data from said installation computer;

c. means for embedding identification data received from said installation computer in said software application to form an identifiable software application;

d. means for generating a cryptographic signature for said identifiable software application; and

e. means for embedding said cryptographic signature in said identifiable software application to form an identifiable and authenticable software application;

said installation computer comprising:

a. an installation computer storage device for storing said identification data, and for storing an installed software application;

b. an installation computer communications interface in communication with said link, for transferring said identification data to said distribution computer and for receiving said identifiable and authenticable software application from said distribution computer; and,

d. means for installing said software application on said installation computer storage device.

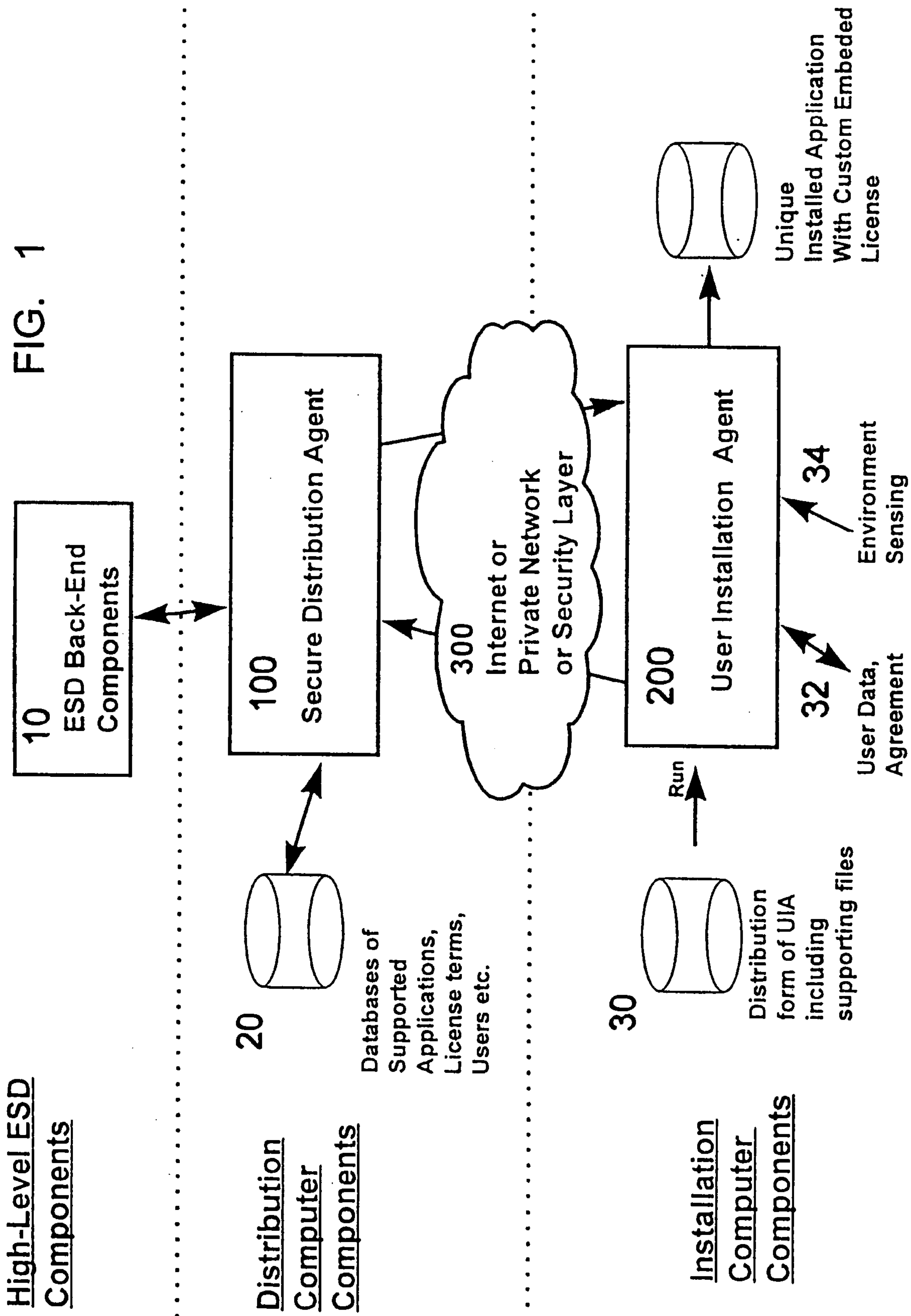

High-Level ESD Components
Distribution Computer Components
Installation Computer Components
10
ESD Back-End Components
100
Secure Distribution Agent
20
Databases of Supported Applications, License terms, Users etc.
300
Internet or Private Network or Security Layer
200
User Installation Agent
30
Distribution form of UIA including supporting files
Run
32
User Data, Agreement
34
Environment Sensing
Unique Installed Application With Custom Embeded License

FIG. 1

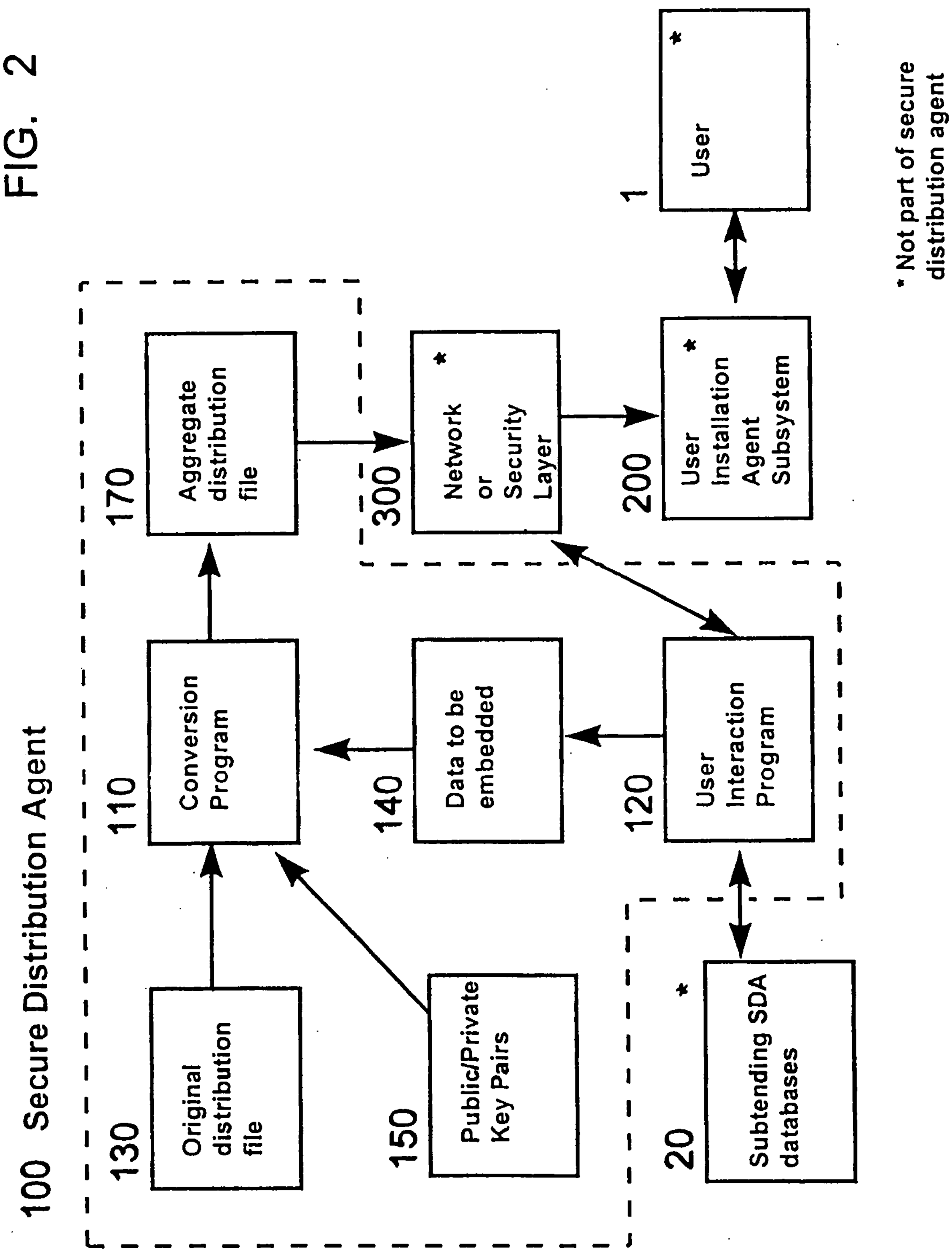

100 Secure Distribution Agent
130
Original distribution file
110
Conversion Program
170
Aggregate distribution file
150
Public/Private Key Pairs
140
Data to be embedded
300
Network or Security Layer
*
20
Subtending SDA databases
*
120
User Interaction Program
200
User Installation Agent Subsystem
*
1
User
*
* Not part of secure distribution agent

FIG. 2

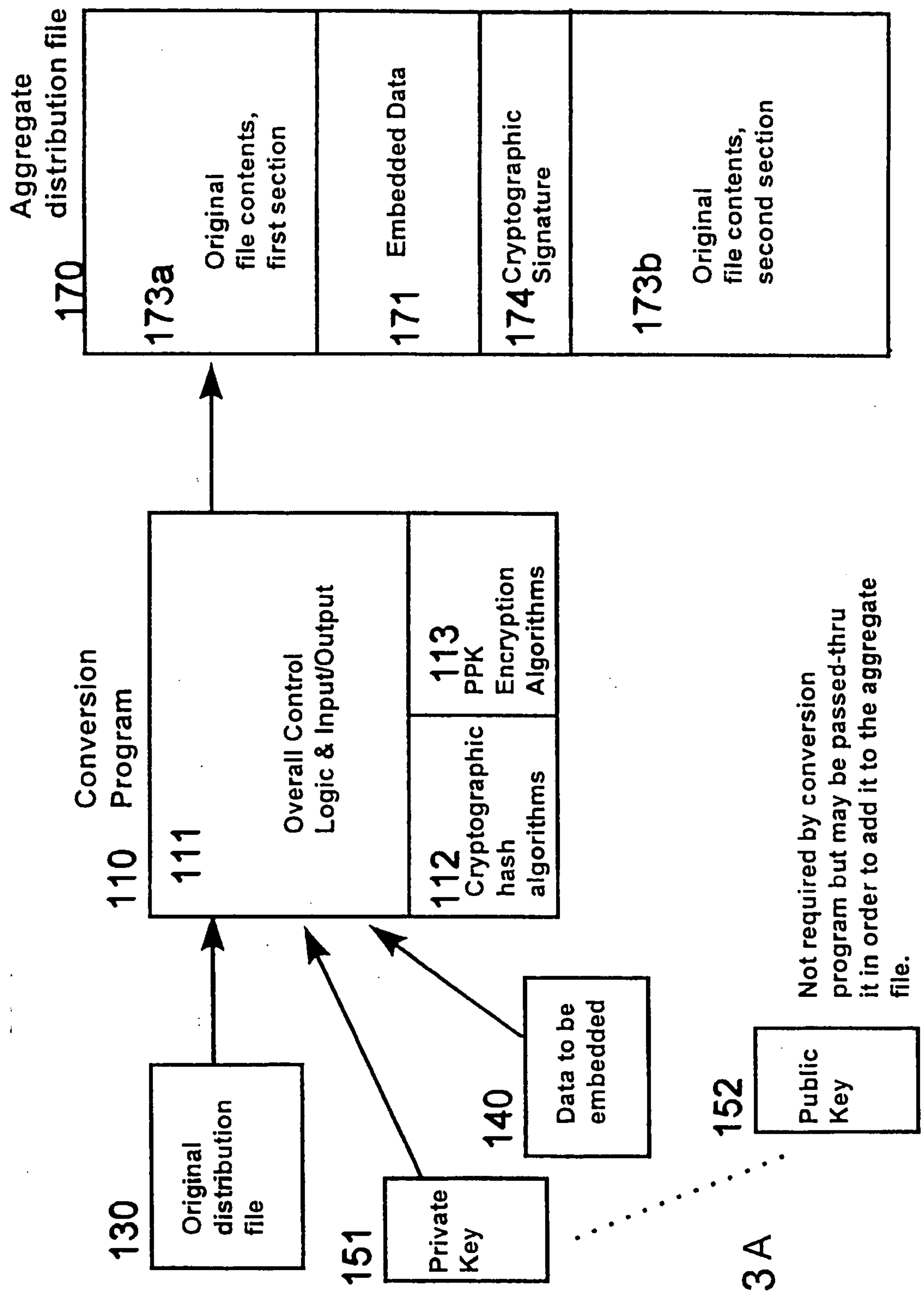

170
Aggregate distribution file
173a
Original file contents, first section
171
Embedded Data
174
Cryptographic Signature
173b
Original file contents, second section
110
Conversion Program
111
Overall Control Logic & Input/Output
112
Cryptographic hash algorithms
113
PPK Encryption Algorithms
130
Original distribution file
151
Private Key
140
Data to be embedded
152
Public Key
Not required by conversion program but may be passed-thru it in order to add it to the aggregate file.

FIG. 3A

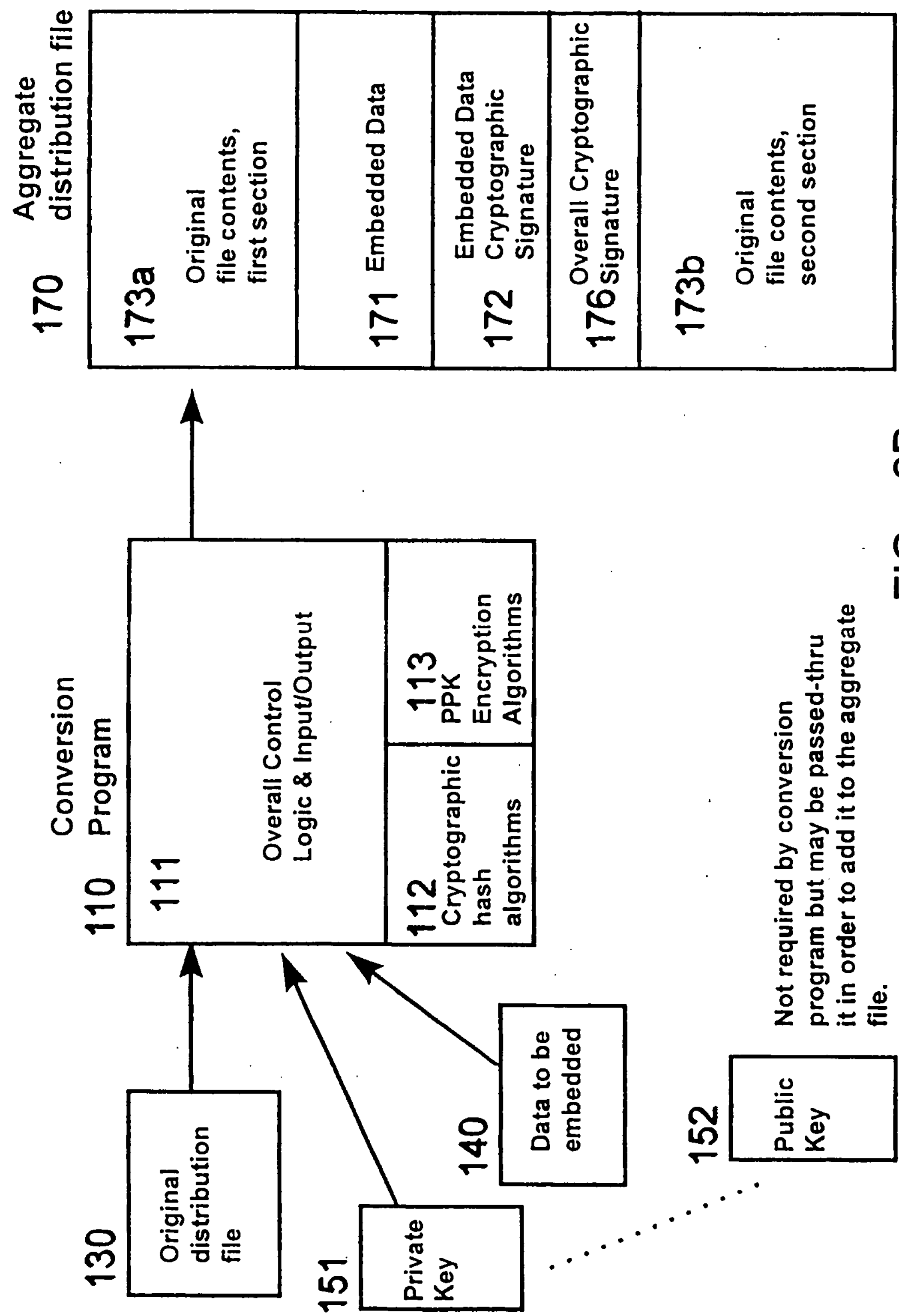
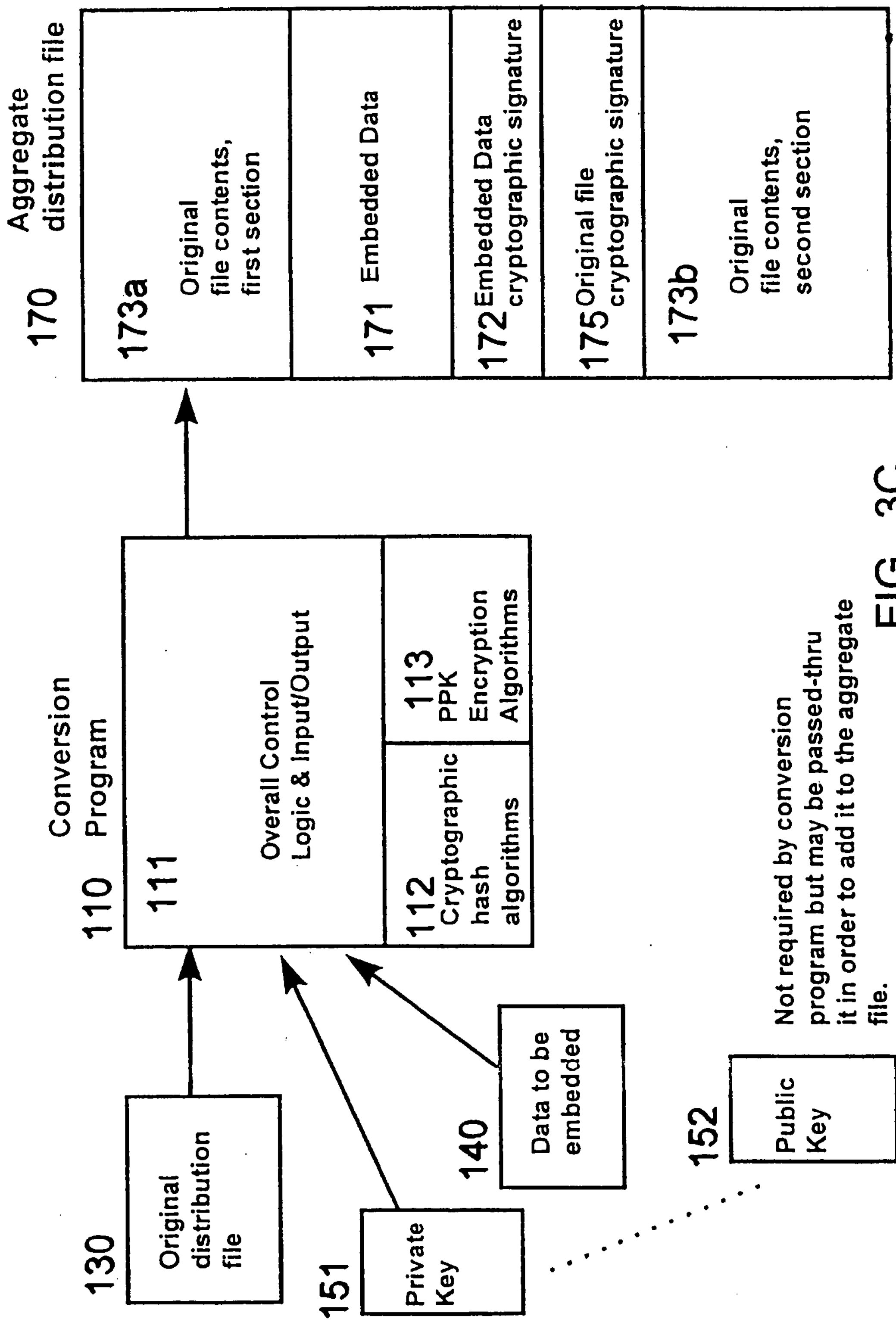

170 Aggregate distribution file
173a Original file contents, first section
171 Embedded Data
172 Embedded Data Cryptographic Signature
176 Overall Cryptographic Signature
173b Original file contents, second section
110 Conversion Program
111 Overall Control Logic & Input/Output
112 Cryptographic hash algorithms
113 PPK Encryption Algorithms
130 Original distribution file
151 Private Key
140 Data to be embedded
152 Public Key
Not required by conversion program but may be passed-thru it in order to add it to the aggregate file.

FIG. 3B



110 Conversion Program
111 Overall Control Logic & Input/Output
112 Cryptographic hash algorithms
113 PPK Encryption Algorithms
130 Original distribution file
151 Private Key
140 Data to be embedded
152 Public Key
Not required by conversion program but may be passed-thru it in order to add it to the aggregate file.
170 Aggregate distribution file
173a Original file contents, first section
171 Embedded Data
172 Embedded Data cryptographic signature
175 Original file cryptographic signature
173b Original file contents, second section

FIG. 3C

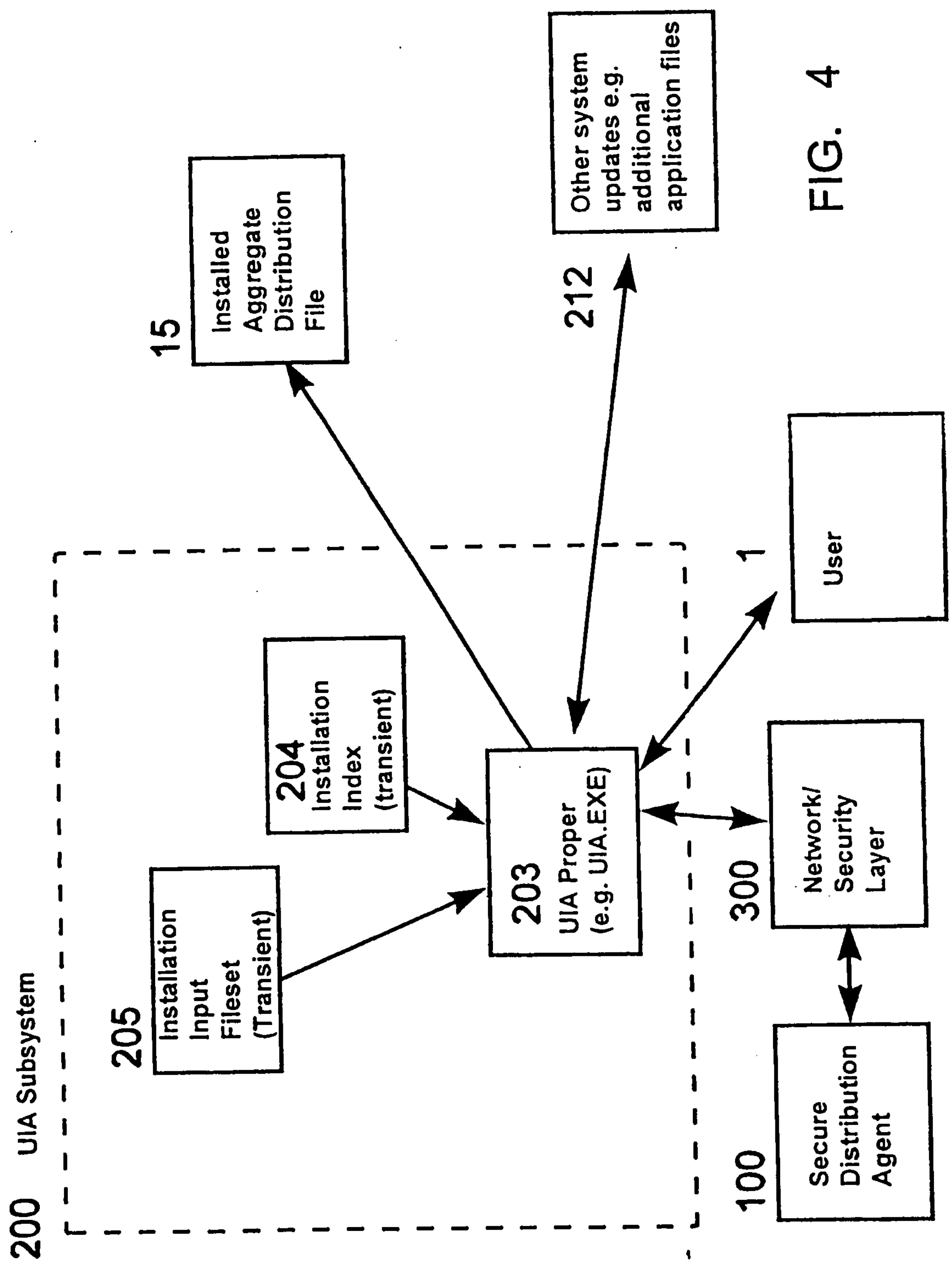

200 UIA Subsystem
205 Installation Input Fileset (Transient)
204 Installation Index (transient)
203 UIA Proper (e.g. UIA.EXE)
15 Installed Aggregate Distribution File
212
Other system updates e.g. additional application files
1
User
300 Network/ Security Layer
100 Secure Distribution Agent

FIG. 4

FIG. 5

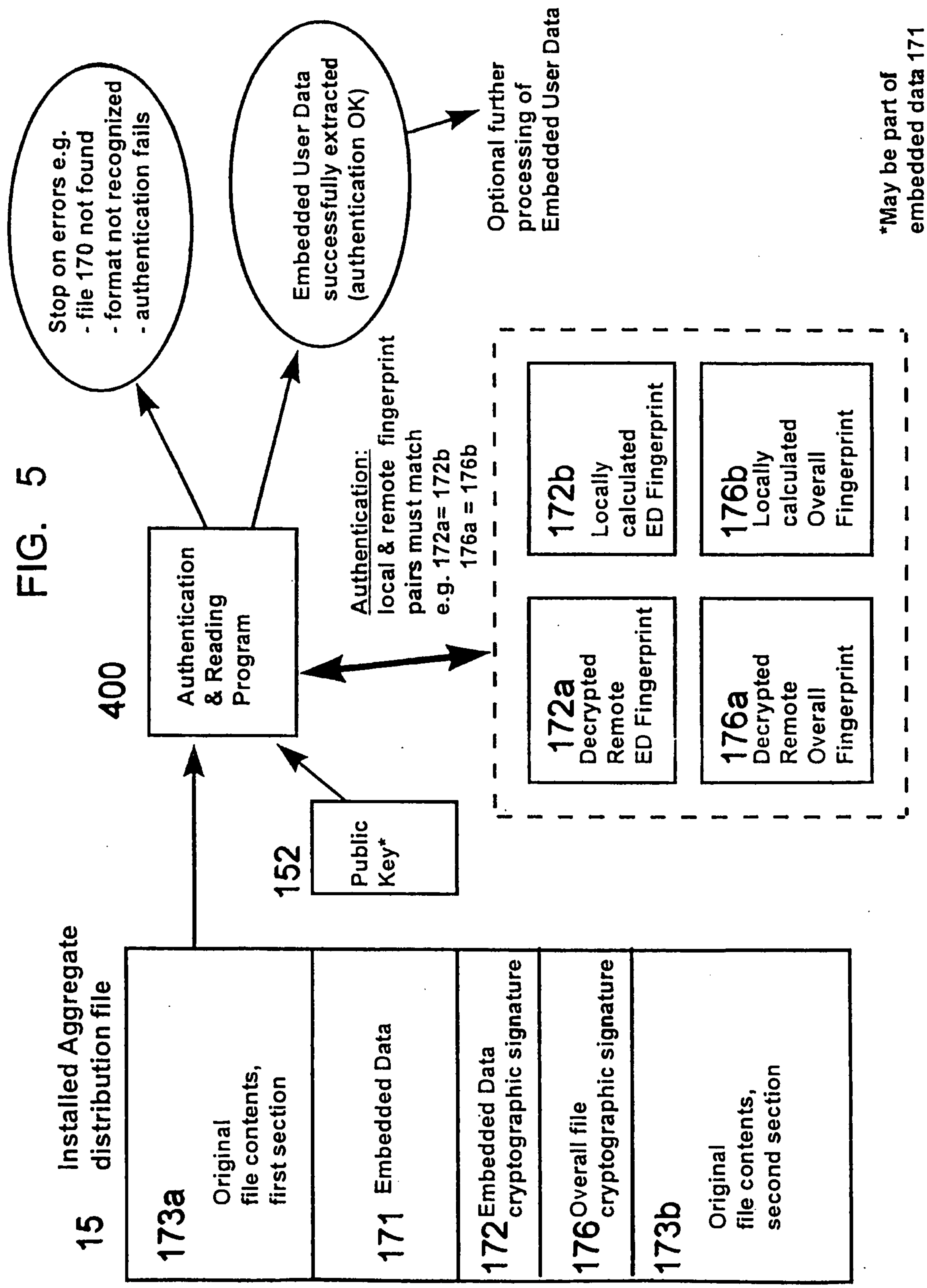
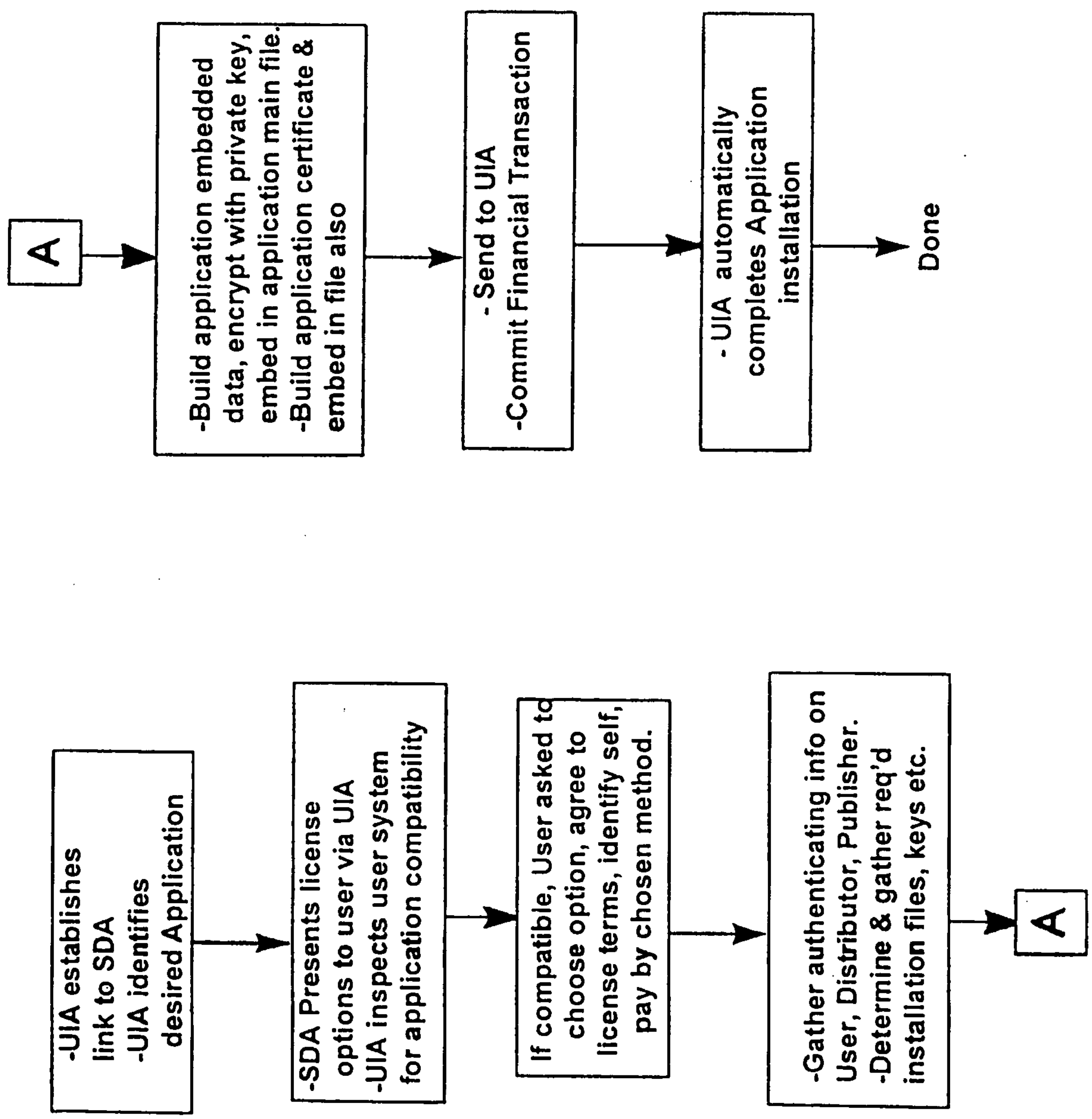
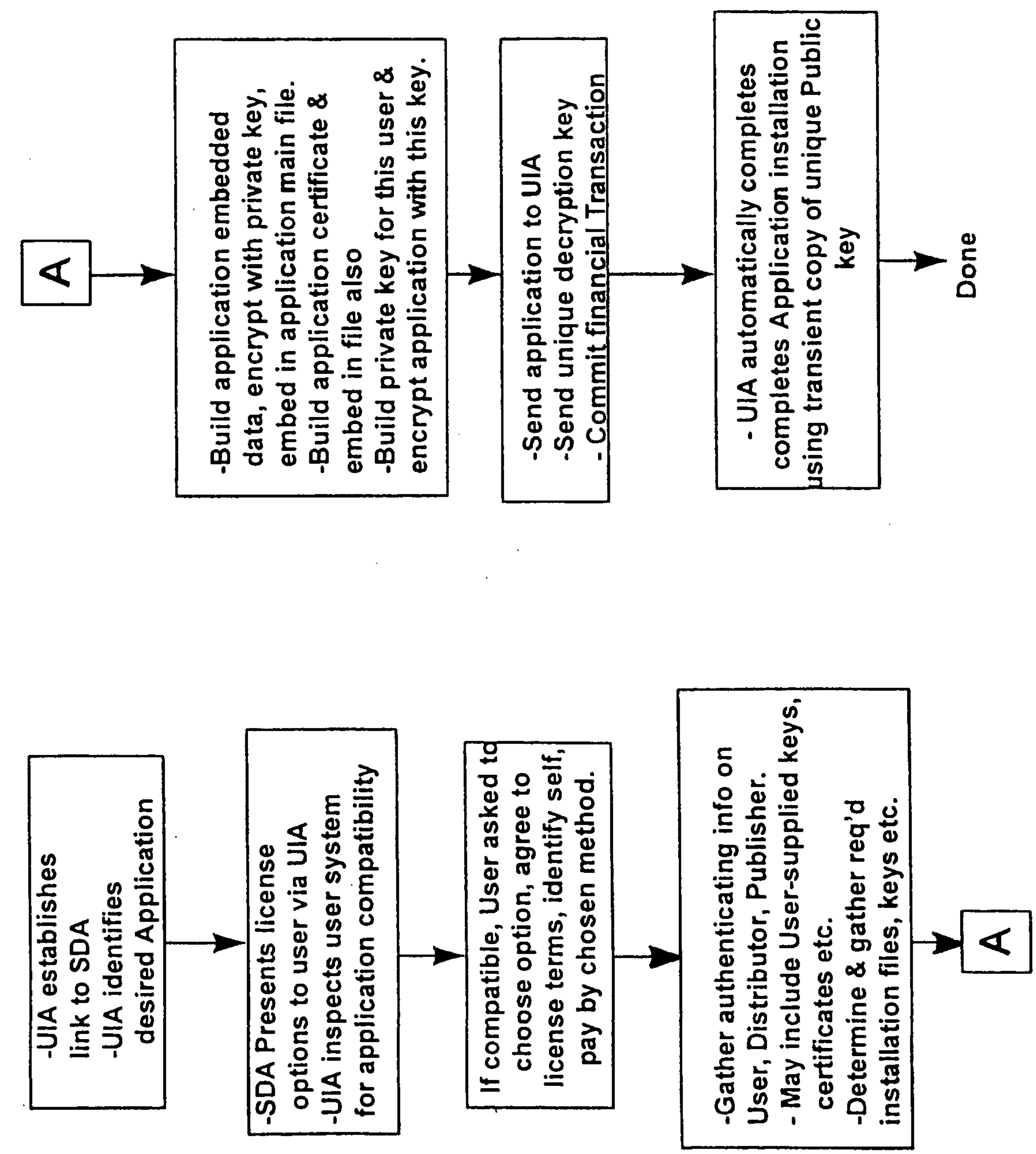

15 Installed Aggregate distribution file
173a Original file contents, first section
171 Embedded Data
172 Embedded Data cryptographic signature
176 Overall file cryptographic signature
173b Original file contents, second section
152 Public Key*
400 Authentication & Reading Program
Authentication: local & remote fingerprint pairs must match e.g. 172a= 172b 176a = 176b
172a Decrypted Remote ED Fingerprint
172b Locally calculated ED Fingerprint
176a Decrypted Remote Overall Fingerprint
176b Locally calculated Overall Fingerprint
Stop on errors e.g. - file 170 not found - format not recognized - authentication fails
Embedded User Data successfully extracted (authentication OK)
Optional further processing of Embedded User Data
*May be part of embedded data 171

FIG. 6


-UIA establishes link to SDA
-UIA identifies desired Application
-SDA Presents license options to user via UIA
-UIA inspects user system for application compatibility
If compatible, User asked to choose option, agree to license terms, identify self, pay by chosen method.
-Gather authenticating info on User, Distributor, Publisher.
-Determine & gather req'd installation files, keys etc.
A
A
-Build application embedded data, encrypt with private key, embed in application main file.
-Build application certificate & embed in file also
- Send to UIA
-Commit Financial Transaction
- UIA automatically completes Application installation
Done

9/10

FIG. 7


-UIA establishes link to SDA
-UIA identifies desired Application
-SDA Presents license options to user via UIA
-UIA inspects user system for application compatibility
If compatible, User asked to choose option, agree to license terms, identify self, pay by chosen method.
-Gather authenticating info on User, Distributor, Publisher.
- May include User-supplied keys, certificates etc.
-Determine & gather req'd installation files, keys etc.
A
A
-Build application embedded data, encrypt with private key, embed in application main file.
-Build application certificate & embed in file also
-Build private key for this user & encrypt application with this key.
-Send application to UIA
-Send unique decryption key
-Commit financial Transaction
- UIA automatically completes completes Application installation using transient copy of unique Public key
Done

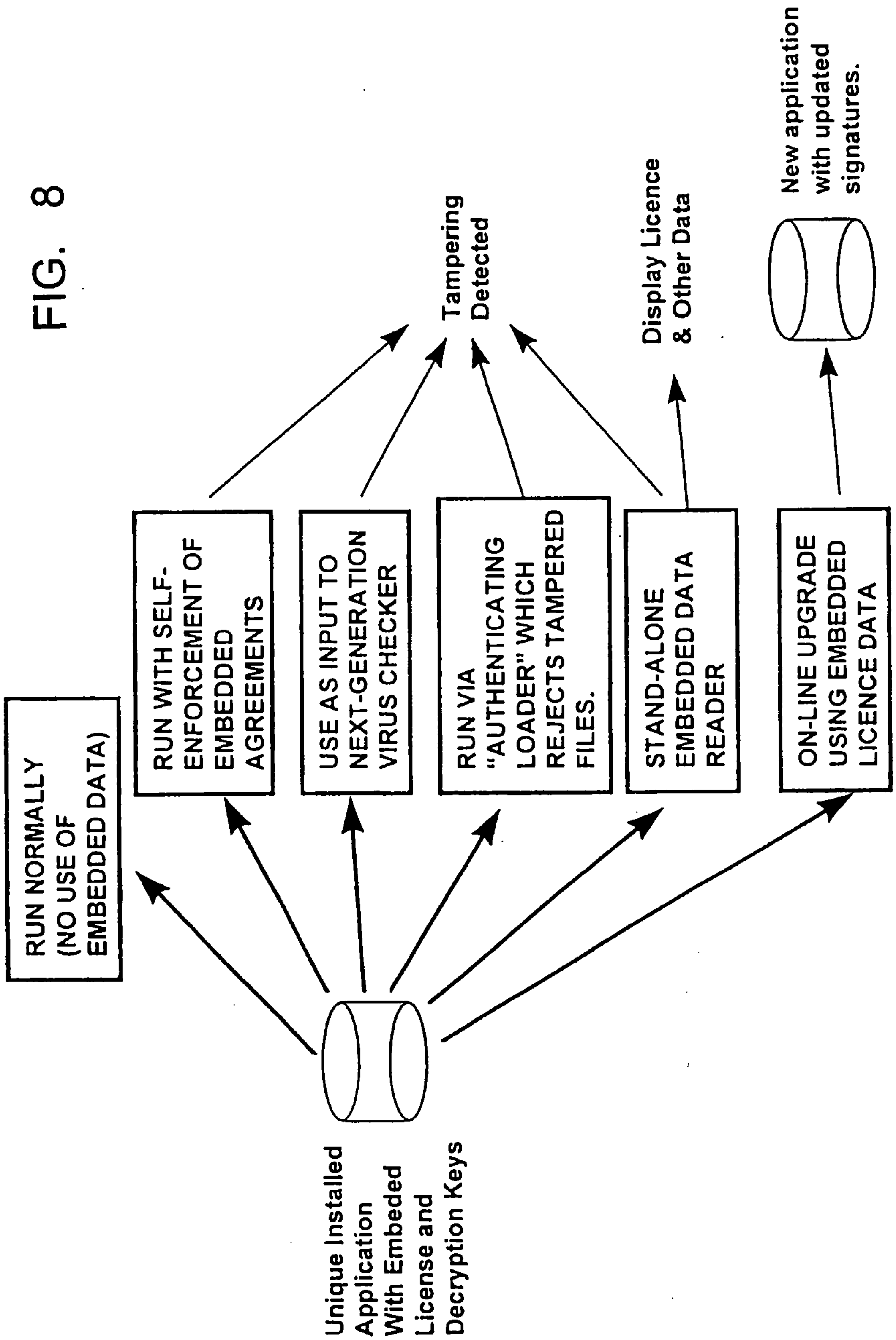

Unique Installed Application With Embeded License and Decryption Keys
RUN NORMALLY (NO USE OF EMBEDDED DATA)
RUN WITH SELF-ENFORCEMENT OF EMBEDDED AGREEMENTS
USE AS INPUT TO NEXT-GENERATION VIRUS CHECKER
RUN VIA "AUTHENTICATING LOADER" WHICH REJECTS TAMPERED FILES.
STAND-ALONE EMBEDDED DATA READER
ON-LINE UPGRADE USING EMBEDDED LICENCE DATA
Tampering Detected
Display Licence & Other Data
New application with updated signatures.

FIG. 8

INTERNATIONAL SEARCH REPORT

Int :ional Application No
PCT/CA 98/00241

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F1/00

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)
IPC 6 G06F

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 0 686 906 A (SUN MICROSYSTEMS INC) 13 December 1995	1-6,8, 10-13, 15-17
A	see column 4, line 1 - column 5, line 5; figures 4,6A,6B ---	7,9,14
A	LEIN HARN ET AL: "A SOFTWARE AUTHENTICATION SYSTEM FOR INFORMATION INTEGRITY" COMPUTERS & SECURITY INTERNATIONAL JOURNAL DEVOTED TO THE STUDY OF TECHNICAL AND FINANCIAL ASPECTS OF COMPUTER SECURITY, vol. 11, no. 8, 1 December 1992, pages 747-752, XP000332279 see page 750, left-hand column, paragraph 2 see page 750, left-hand column, paragraph 8 - right-hand column, paragraph 5 --- -/--	1-17

[X] Further documents are listed in the continuation of box C.

[X] 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	Date of mailing of the international search report
8 July 1998	15/07/1998

Name and mailing address of the ISA	Authorized officer
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	Moens, R

1

INTERNATIONAL SEARCH REPORT

Int   'onal Application No
PCT/CA 98/00241

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 509 074 A (CHOUDHURY ABHIJIT K ET AL) 16 April 1996 cited in the application see abstract ---	1-17
A	EP 0 717 337 A (IBM) 19 June 1996 see column 1, line 1 - column 2, line 40 -----	1,8,12, 15-17

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/CA 98/00241

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0686906 A	13-12-1995	US 5724425 A	03-03-1998
		JP 8166879 A	25-06-1996
US 5509074 A	16-04-1996	CA 2137065 A	28-07-1995
		EP 0665486 A	02-08-1995
		JP 7239828 A	12-09-1995
EP 0717337 A	19-06-1996	JP 8221268 A	30-08-1996
		US 5745678 A	28-04-1998

Form PCT/ISA/210 (patent family annex) (July 1992)

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04L 9/08

A1

(11) International Publication Number: **WO 99/34553**

(43) International Publication Date: 8 July 1999 (08.07.99)

(21) International Application Number: PCT/US98/27531

(22) International Filing Date: 30 December 1998 (30.12.98)

(30) Priority Data:
09/001,463 31 December 1997 (31.12.97) US

(71) Applicant: V–ONE CORPORATION [US/US]; Suite 300, 20250 Century Boulevard, Germantown, MD 20874 (US).

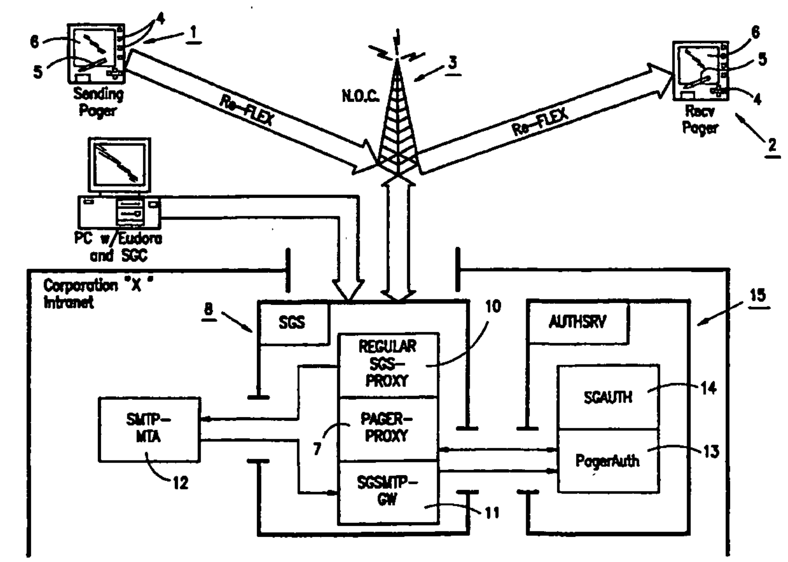
(72) Inventors: WRIGHT, Steven, R.; Apartment 21, 12010 Waterside View Drive, Reston, VA 20194 (US). BROOK, Christopher, T.; 7308 Pomander Lane, Chevy Chase, MD 20815 (US).

(74) Agents: URCIA, Benjamin, E. et al.; Bacon & Thomas, PLLC, 4th floor, 625 Slaters Lane, Alexandria, VA 22314 (US).

(81) Designated States: 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, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, 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.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: KEY ENCRYPTION SYSTEM AND METHOD, PAGER UNIT, AND PAGER PROXY FOR A TWO–WAY ALPHANUMERIC PAGER NETWORK



(57) Abstract

A method and system allows encryption services to be added to an existing wireless two–way alphanumeric pager (4) network by providing a pager proxy (7) which is arranged to receive an encrypted message from a sending pager (1) and re–packages it for retransmission to the destination pager (2). The sending pager encrypts the message using a session key, and encrypts the session key so that it can only be recovered by a secret key of the pager proxy. Authentication (13) of the sending pager and proxy server is provided by encryption of the session keys together with identifying data, and authentication of the message is provided by a message authentication code generated by computing a message authentication code based on the session key, identifying data, and the message.

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
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

KEY ENCRYPTION SYSTEM AND METHOD, PAGER UNIT, AND PAGER PROXY FOR A TWO-WAY ALPHANUMERIC PAGER NETWORK

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to a system and method of encrypting messages for transmission and/or receipt by a pager, and in particular to a system and method for which uses a standard two-way wireless pager protocol to send encrypted messages over an existing paging infrastructure. The invention also relates to a pager unit capable of sending and receiving encrypted alphanumeric messages over a wireless pager network, and to a pager proxy server which provides key management functions for enabling transmission of encrypted alphanumeric messages over the wireless pager network.

2. Description of Related Art

Paging systems capable of transmitting simple alphanumeric messages and displaying the messages on a miniature two-way pager are becoming increasingly popular. Such two-way paging systems enable messages like "Meet me at the gym at 6:00" or "I love you" to be both transmitted and received by equipment that is smaller, less complex, and less intrusive than a wireless telephone. The messages are transmitted as packets containing source and destination address data formatted for transmission over the response channel of a wireless paging network, using a protocol that allows users to respond to messages directly from their pager units without having to use a telephone.

Two-way pagers are currently offered by Motorola and Wireless Access, with national paging services being provided by MTEL, which uses Motorola's Re-FLEX™ paging protocol. The Re-FLEX™ paging protocol allows users to respond to messages using a selection of pre-programmed responses or by formatting a free-form text reply, and in addition includes a TCP/IP protocol stack that allows the user to initiate messages to subscribers on wired networks, including e-mail and fax machine addresses.

The present invention concerns a method and system for encrypting and authenticating messages transmitted over the existing pager system, using the Re-FLEX™ protocol, or over other yet-to-be-implemented paging systems in the U.S. and elsewhere which may or may not use the Re-FLEX™ protocol. Unlike previously proposed arrangements, which either rely on complex encoding schemes and sophisticated hardware at the sending and destination ends of a transmission, over transfer of keys and authentication of keys using a telephone rather than the wireless network, the present invention offers the advantages of (i) providing authenticable key encryption of messages at the source of the transmission and key decryption at the destination, with protection of the communication by keys that are unique to each pager, rather than shared, and yet with no need for a key exchange between the sending and destination pagers, (ii) using existing two-way pager designs and paging system infrastructure, and (iii) providing the encryption capabilities without adding to carrier overhead. The addition of full key encryption and authentication capabilities to an existing pager system without adding to carrier overhead or capital costs distinguishes the system and method of the invention from all previously proposed pager encryption schemes.

An example of a previously proposed pager encryption scheme is described in

U.S. Patent Nos. 5,452,356 and 5,481,255, assigned to Data Critical Corp. Although the term "encryption" is used in these patents, the patents are directed primarily to a data compression and encoding protocol for enabling transmission of large volumes of data over a wireless pager network using modified transmitting and receiving hardware, including separate computers at each end of the transmission. The only discussion of encryption in these patents is a cursory reference to "encryption" as an added security layer provided by utilizing a "commercially available algorithm" (see, *e.g.*, col. 11, lines 15-32 of U.S. Patent No. 5,452,356) during encoding of the files by a computer connected to the pager. Because all encryption and decryption in the Data Critical patents is disclosed as being carried out by software on computers connected directly to the sending and receiving pagers, the only possible ways that true key encryption could be provided for would be to use encryption keys corresponding to decryption keys common to all possible recipients of the message, to use unique keys for each potential recipient but to store the corresponding encryption keys in the sender's computer, or to exchange keys prior to a transmission. While these alternatives might be reasonable in the context of, for example, a medical paging system in which all transmissions are between doctors or trusted medical personnel, none of them are suitable for use in connection with a paging system designed to transmit simple text messages using miniature handheld paging units and which is open to the general public, both because of the hardware intensive nature of the encoding scheme and the problem of key management.

In addition to the wireless pager protocol described in the Data Critical patents the prior art includes a number of patents describing authentication or encryption schemes that are used in connection with wireless paging, but are carried out over a telephone line. The systems described in these patents are more suited to traditional one-way paging environments than with two-way protocols, even though one of the patents issued only recently, and none disclose systems that can be practically applied to the current two-way paging networks.

U.S. Patent No. 5,668,876, for example, discloses a modified pager which provides authentication of a caller. The modified pager calculates a unique response code based on a transmitted challenge code, an input personal identification number, and an internal key. The resulting response code is converted into DTMF tones and transmitted by telephone to a central computer which authenticates the caller. This system does not provide for encryption of messages, or authentication by the receiving party of communications forwarded by the central computer, and yet requires a challenge response form of authentication which requires simultaneous two-way communications, which is currently neither possible nor required by existing two-way wireless pager protocols.

U.S. Patent No. 5,285,496 describes a paging system with two options: the first is to send and receive encrypted messages using private key encryption by transmitting a clear text message over a private communications line to a local client of the pager network where the message is encrypted using a private key, and broadcast over a pager network, and the second is to send the message in clear text by telephone directly to the central control system of the pager network, where the message is encrypted. However, neither of the two options provides for encryption of the original pager transmission, which must be sent in clear text form over a telephone line, and which, in the case where a local client computer is used, provides no way to maintain centralized control. In addition, for the local client computer option, in which the address is encrypted together with the message, the destination pager must decrypt every message sent over the system in order to determine whether a message is addressed to it, which is only possible in pager networks with a very limited number of participants.

In the system described in U.S. Patent No. 5,638,450, on the other hand, reception by a pager of encrypted messages over a radio frequency pager network is made possible by having the pager transmit an encryption key via DTMF tones over a telephone line to a central office, the central office then encrypting the messages before forwarding them to the recipient. This system does not permit outgoing messages to be encrypted, and provides no way of key encrypting messages between two pagers on the network, and again is not applicable in the context of the present invention.

It will be appreciated that none of the above patents, representing the known pager message protection proposals, describes a system that enables true key encryption and authentication capabilities to be added to a conventional two-way wireless alphanumeric paging system of the type with which the present invention is concerned, using existing pager protocols and equipment, and in which any individual can send a simply alphanumeric message by keying the message into a miniature two-way pager (or choosing from a menu of pre-stored messages), entering a destination address, and pressing a send button, the message then being retrievable by the intended recipient by a simple keystroke on the recipient's pager, with the message being encrypted by a key unique to the sending pager and decrypted by a key unique to the destination pager. In contrast, the present invention not only provides these capabilities, but adds further levels of security by using strong secret or private key based encryption algorithms, with multi-tier authentication of a transmitted packet, while permitting central registration and billing for encryption services and recovery of messages by legal authorities without adding to carrier overhead or increasing the costs of the paging service for users who do not require encryption.

All of the above advantages of the system and method of the invention are made possible through the use of a proxy server to intercept an encrypted message and repackage it for delivery to the intended recipient in a form that the intended recipient is capable of reading, thus eliminating the need for shared keys or key exchange between the sender and ultimate recipient of the message or complex, hardware-intensive encoding schemes, and allowing encrypted messages to be transmitted using existing two-way alphanumeric pager protocols. Because the invention involves key encryption and not encoding of the message, and requires knowledge by the sending and receiving units of only one or two keys (for example, a private key unique to the pager and a server's public key), encryption being simpler to implement than encoding since it merely involves performing arithmetically combining the message with the key, the present invention can be used with existing pager hardware and protocols, and by avoiding the need for challenge/response authentication, the present invention can be used with existing channels and therefore with the existing pager infrastructure. None of the previously proposed systems and methods has these capabilities.

Not only does the use of a proxy server relieve the sending and receiving pagers of key management functions, but the manner in which the invention utilizes strong encryption capabilities, by separately encrypting the session key, further minimizes the processing resources required by the sending and receiving pagers. Essentially, encryption of the message itself can be carried out with a relatively short session key to minimize usage of the processor, while the relatively short session key can be protected by a strong encryption algorithm. Because the session key is not re-used, key integrity can easily be maintained.

SUMMARY OF THE INVENTION

It is accordingly a first objective of the invention to provide a system of adding full key encryption services to a pager network, allowing key encrypted alphanumeric messages to be sent by any pager unit registered with the encryption service provider to any other registered pager unit via the network, as well as to e-mail addresses, fax machines and other destinations capable of receiving text messages.

It is a second objective of the invention to provide a method of adding full key encryption services to a pager network, allowing key encrypted messages to be sent by any pager unit registered with the encryption service provider to any other registered pager unit via the network, as well as e-mail addresses, fax machines and other destinations capable of receiving text messages.

It is a third objective of the invention to provide a system which allows encryption of alphanumeric messages by a paging unit for wireless transmission over a paging network in a manner which is transparent to the person sending the message, and which allows decryption and display of the messages by a receiving pager in a manner which is transparent to the person receiving the message.

It is a fourth objective of the invention to provide a method which allows encryption of messages by a paging unit for wireless transmission over a paging network in a manner which is transparent to the person sending the message, and which allows decryption and display of the messages by a receiving pager in a manner which is transparent to the person receiving the message.

It is a fifth objective of the invention to provide a system and method of adding encryption capabilities with centralized key management and unique secret keys for each user, without modification of existing pager network infrastructure or paging transmission protocols.

It is a sixth objective of the invention to provide a system and method of encrypting text messages capable of being transmitted over a pager network, which can be provided as an add-on or option to the services provided by the pager network, and which can be centrally managed using a proxy server connected to the network to provide the encryption services to subscribers who select the encryption option.

It is a seventh objective of the invention to provide a system and method of authenticating messages transmitted in encrypted form over a pager network, without the need for an authentication channel or challenge/response protocol.

It is an eighth objective of the invention to providing a standard alphanumeric pager unit with the capability of encrypting, decrypting, and authenticating messages transmitted using a two-way alphanumeric pager protocol, with minimal or no hardware modification.

It is a ninth objective of the invention to provide a proxy server arrangement which can be connected to the network operations center of a pager network in order to manage transmission of key encrypted messages over the network.

These objectives are achieved, in accordance with the principles of a preferred embodiment of the invention, by using a pager proxy server to carry out decryption of a message encrypted by a session key and received from the sending pager, and to have the pager proxy generate a new session key for re-encryption of the message transmitted to the receiving pager, with the original session key being encrypted at least by a secret key shared by the sending pager and the pager proxy server or by a public key corresponding to a private key of the pager proxy server, and the new session key being encrypted either by a secret key shared by the pager proxy server and the destination pager or a public key corresponding to a private key held by the destination pager, thereby freeing the sending and destination pagers from having to store more than one secret key or of having to carry out a direct exchange of keys, and allowing each pager on the network to be provided with a unique key.

In accordance with the principles of an especially preferred embodiment of the invention, in order to encrypt a message, the sending pager must have hard-coded into memory a unique identification number and a secret key associated with the identification number. When a user is ready to send an encrypted message, he or she begins by entering the message to be sent, after which the user is prompted for an access code to gain access to the encrypted shared key, the encrypted shared is decrypted, and a session key is generated. The message that was entered by the user is then encrypted with the session key, and the session key is encrypted with the public key of the pager proxy server, or a shared secret key of the sending pager, and appended to the encrypted message for transmission via the network operations center to the pager proxy server.

Pager messages are formatted in accordance with standard pager protocols to include a destination header, which is generally the address or telephone number of the receiving pager, and with an additional space in the header to indicate that the message is encrypted, as will be explained in more detail below. When the network operations center receives a message that is in encrypted form, it forwards it to the encryption service center, which must at least include a pager proxy server, using an appropriate protocol, examples of which include but are not limited to TME-X and TNPP. In the illustrated embodiment, the pager proxy server is included in a gateway server in order to enable the system to package e-mail messages for transmission in encrypted form to pagers on the pager network, or to package pager messages according to an e-mail protocol for transmission over a wired network such as the Internet to an e-mail address, but it will be understood by those skilled in the art that the pager proxy may be operated as a separate unit.

In the illustrated embodiment of the invention, the pager proxy server has the role of verifying the authenticity of the message sent by the sending pager, decrypting the data with its private key or alternatively with a secret key shared with the sending pager to obtain the session key that was generated by the sending pager, and decrypting the message with the session key generated by the sending pager. Once this is accomplished, the server generates a new session key to encrypt the message with, and then encrypts the session key with a secret key shared with the destination pager or with a public key corresponding to the private key of the destination pager, or alternatively with a secret key shared with the destination pager, the two entities being appended together and sent to the recipient pager. The destination pager, after receiving the encrypted message, alerts the user and, when the user is ready to read the encrypted page, prompts him or her for the access code to begin decryption of the appropriate shared secret key or private key, which is then used to decrypt the session key used to decrypt the message.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic diagram showing the principal elements of a pager encryption system constructed according to the principles of a preferred embodiment of the invention.

Fig. 2 is a schematic illustration summarizing the operation of the two-way pager for sending an encrypted message over a wireless network in accordance with the principles of a preferred embodiment of the invention.

Fig. 3 is a functional block diagram of a module used by a two-way pager to encrypt a message and package it for wireless transmission over a pager network to a network operations center.

Fig. 4 is a functional block diagram of a module used by a pager proxy server to authenticate the sender of an encrypted message, authenticate the message, and extract information from the message which can be used to re-package the message for transmission a destination address.

Fig. 5 is a functional block diagram of a module used by the pager proxy server to repackage a message and send it to the network operations center for transmission for re-transmission over the wireless pager network to a destination pager.

Fig. 6 is a functional block diagram showing the principal elements of a module used by a destination pager to decrypt and display a message received in encrypted form from the network operations center over the wireless paging network.

Fig. 7 is a flowchart of a preferred process corresponding to the functional block diagram of Fig. 3.

Fig. 8 is a flowchart of a preferred process corresponding to the functional block diagram of Fig. 4.

Fig. 9 is a flowchart of a preferred process corresponding to the functional block diagram of Fig. 5.

Fig. 10 is a flowchart of a preferred process corresponding to the functional block diagram of Fig. 6.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

As illustrated in Fig. 1, the system of the preferred embodiment of the invention allows encrypted communications between a sending pager and a receiving pager via a two-way wireless paging system such as M-TEL's system, using two-way alphanumeric pagers such as, but not limited to, the Motorola and Wireless Access pagers. The basic elements of the system are a sending pager 1, a receiving pager 2 which may be identical to the sending pager, and a network operations center (NOC) 3 which provides basic message forwarding and subscription management services for all communications carried by the system.

As is conventional, the sending and receiving or destination pagers (or pager units) 1 and 2 include function and data entry keys 4, and/or a stylus 5 or other data entry device, for allowing a user to input and send alphanumeric messages, and an LCD or other device 6 which allows received alphanumeric messages to be displayed. The pagers can also provide other functions such an alarm function to alert the user that a message has been received, and includes a microprocessor and circuitry capable of formatting an input message and transmitting it to the network operations center according to an appropriate protocols, including but not limited to the ReFLEX™ protocol. The sending and receiving or destination pagers also include a memory for storing a unique user identification number (UID) that identifies a particular pager for addressing purposes, and other information such as a password that can be used to prevent unauthorized users from accessing the transmission or message display functions of the pager, as well as an addressing mode (AM) generator that is used in the pager protocol to indicate the type of addressing used by the paging system, and a timer that can be used to generate a message number.

In order to be used with the system and method of the illustrated embodiment of the invention, the pager memory must also have stored therein at least a private key of the pager unit, a corresponding public key of the pager unit, and a public key corresponding to a private key of the server, for encrypting either the message itself or a session key used to encrypt the message, and software capable of running on the included processor for performing an encryption algorithm and a decryption algorithm. In addition, according to the preferred embodiment of the invention illustrated in Figs. 2-10, the pager must be capable of generating a session key for each message to be transmitted, storing a private key unique to the pager which is used to authenticate the pager, and computing a message authentication code which is used to authenticate the message being transmitted or received.

It will be appreciated by those skilled in the art, however, that whenever a public key or private key is required, a shared secret key could be substituted using an appropriate algorithm, and that while the use of session keys is highly advantageous, the session key could also be eliminated in favor of public-private key encryption. In addition, while the illustrated system provides both encryption and decryption capabilities in at least two pagers, so that each pager can send or receive messages, the system and method of the invention could also be applied to systems in which some or all of the pagers have reception capabilities only, *i.e.*, in which some or all of the pagers are designed to allow the pagers to receive encrypted messages originating from e-mail addresses and/or two-way pagers, but not to originate messages. Conceivably, the system and method of the invention could even be applied to systems in which at least some of the pagers are capable of sending encrypted messages, but not receiving and decrypting them, although such a system would seem to make little commercial sense. In any case, it will be appreciated that the system and method illustrated in Figs. 2-10 are intended as being illustrative in nature only, and should not be interpreted as being limitative of the scope of the invention.

As indicated above, the number of keys required of a pager to encrypt and decrypt messages is at most two, so that the key storage requirements are minimal. The encryption algorithms themselves simply involve a series of mathematical steps, and are well within the capabilities of the microprocessors used in the conventional pagers, as are message authentication code generating techniques such as CRC or SHA1. The session key used in the preferred embodiment to encrypt the message itself consists, in a practical implementation, of just sixteen characters (128 bits), and thus encryption of the alphanumeric message using RC4 or a similar stream cipher or other algorithm which makes use of a shared secret key can be accomplished without a large amount of processing resources, while strong overall protection of the transmission is still provided because the more processor intensive encryption algorithms are reserved for encryption of the relatively small session key rather than the alphanumeric message itself. Of course, the session key is not limited to a particular bit size, and it is possible for example to use 256 bit session keys, or longer or shorter session keys as desired.

In the preferred embodiment, encryption of the session key is carried out by RSA (1024 bits) but other stronger private key algorithms such as ECC PK1 (~2500 bits) can also be used, as well as shared secret key-based encryption methods such as RC4: The public-private key encryption algorithms are preferred not only because of the strong encryption provided, but also because the permit authentication of the sender, as explained below, but legal or other considerations may also affect the choice of encryption algorithm, and thus the system of the invention is designed to permit the use of different mutually exclusive encryption algorithms by the sending and destination pagers.

The sending pager 1 illustrated in Fig. 1 transmits messages to the network operations center 3 in the form of a packet that includes a clear text applications header that tells the center to forward the text to the pager proxy server 7, which is conveniently though not essentially included in a gateway 8 capable of network communications as well as the pager encryption and decryption functions required by the present invention. Forwarding of the packet to the pager proxy or gateway server preferably involves use of a network data transfer protocol such as TME-X, although the manner in which the packet is forwarded to the proxy will depend on the wireless protocol used by the pager network and the capabilities of the network operations center. TME-X is a preferred transfer protocol for use with Re-FLEX encoded packets because of the presence of a TCP/IP stack in the standard format packets that allows the Re-FLEX™ protocol to communicate directly with computer networks.

The gateway 8 may include a general purpose proxy server 10 such as the one described in U.S. Patent No. 5,602,918, entitled "Application Level Security System And Method," and also in U.S. Patent Application Ser. No. 08/917,341, filed August 26, 1997, entitled "Multi-Access Virtual Private Network," both of which are incorporated herein by reference. The two patent documents describe a system currently available from V-One Corporation of Germantown, Maryland under the name SmartGate™ (SG in the figures) which is especially suitable for use with the pager proxy of the present invention, although the pager proxy server of the invention could also be used with other gateway servers, or without any network connection capabilities.

As illustrated, gateway 8 also includes a dedicated e-mail server or gateway 11, and e-mail protocol message transfer agent (MTA) 12 for transferring messages from the gateway server 10 to the e-mail gateway. Both the e-mail gateway 11 and pager proxy 7 may be physically incorporated in the gateway server or provided on independent or separate computers, and are connected to a pager authentication module 13 which may be physically incorporated into a general purpose gateway authentication module 14 of a separate authentication server 15, combined with the gateway server, or may be provided as an independent unit.

Computers on the network with capabilities of communicating with the general purpose proxy server are represented in Fig. 1 by computer 16, and include gateway client software that permits the computer to establish a secured communications path to the gateway server, as well as an e-mail program which packages messages in an appropriate format such as that provided by the SMTP protocol for transmission over the secured communications path established by the gateway client software. An example of an e-mail program is "Eudora™," although the use of standard protocols such as SMTP and Re-FLEX™ allows any e-mail program to communicate with the gateway and thence with the pager network, so that the system of the invention is not limited to use in connection with any particular e-mail program, the conventional pager network already being equipped to handle e-mail transmissions to or from the wireless network. The invention may be considered to apply equally to pager-to-pager communications, pager-to-email communications, and email-to-pager communications. In addition, it is possible that the invention could be adapted to communications originating from a fax machine, in which case the clear packet transmitted by the fax machine over a telephone line would be processed by a facsimile proxy for packaging and encryption by the pager proxy, and messages addressed to the fax machine would be decrypted by the pager proxy and forwarded to the facsimile proxy for transmission as clear text over a telephone line, the principles of the invention still being applicable to the encryption and decryption of the messages by the pager proxy and sending or receiving pagers.

Turning to the specific embodiment illustrated in Figs. 2-10, the system and method of the invention take the form of modifications to the header of the transmission packet sent by the sending pager 1 and/or the pager proxy 7. Essentially in order to send messages over the paging system, the sending pager and pager proxy, (or pager proxy alone in the case of a message originating from computer 16 or a source of clear text messages such as a facsimile machine) generates a header which includes the information necessary to enable processing by the recipient of the packet, and in the case of the pager proxy, for forwarding of a repackaged packet to a destination address. The header should at least include the session key encrypted message, the encrypted session key, a sender identification number, and a destination header or address, but because the header format will vary if a protocol other than Re-FLEX™ is used, it should be appreciated that the other information contained in the illustrated header, and the position of the information, can be varied without departing from the scope of the invention, and the invention is intended to encompass headers formatted for other alphanumeric wireless paging protocols, as well as for encryption algorithms and authentication protocols other than the specific algorithms and protocols indicated.

Fig. 2 illustrates the format of the preferred header, which is divided into three fields. It is to be understood that while the illustration refers to the communication between the sending pager and the pager proxy, the same header will be used for the communication between the pager proxy and the destination pager, with appropriate substitutions of addresses and keys as explained in more detail below. As shown in Fig. 2, the first field is a clear text field that contains the encryption method indicator EM, pager addressing mode (AM), and user identification number (UID) (sometimes referred to as a PIN, but not to be confused with the password entered by the user to access pager functions), while the second field contains the encrypted session key (SESKey1) and various data referred to as "header data" (HdrData) including the destination header or address (DH) and a message authentication code (MAC), the information in the second field being encrypted by the unique private key of the sending pager (pv.sender) in order to authenticate the sender, and by a public key corresponding to a private key held by the server (pb.server) in order to protect the contents of this field. The third field contains the message encrypted by the session key.

The various fields illustrated in Fig. 2 may be formatted in any convenient manner permitted or required by the protocol used to package the data in the fields for transmission, but in the illustrated example most or all of the data in at least fields one and two can conveniently be in hexadecimal format. Whenever the drawings illustrate a hexadecimal number, the number ## will be preceded by a "0x" to form 0x##.

The encryption method indicator EM indicates which of the possible encryption methods handled by the server is being used to encrypt the session key and other information in field 2, so that the session key can be recovered and used to decrypt the encrypted message in field 3. As indicated above, possible encryption methods include the RC4 secret key encryption method, which requires the parties to the communication to have a shared secret key that is used for both encryption or decryption, and the RSA public key encryption method, which is the method illustrated in Fig. 2. The indicator itself is simply a number assigned to the encryption method. While any given pager will generally have only a single encryption method stored in memory, it is possible for the pager proxy to be arranged to handle multiple different methods and thus need to have an indication of the type of encryption method, to accommodate different pager systems or legal requirements, particularly if international pager traffic is involved.

The addressing mode (AM) indicates the type of address involved. For example, in the U.S., pager addressing modes are assigned one application header, while e-mail addressing modes are assigned another application header. This indicator may not be necessary in all protocols since the destination header may be unique to a specific type of address, but is included in field 1 as part of the Re-FLEX™ protocol.

The user identification number (UID) included in clear text in field 1 and in encrypted form in field 2, is the unique address assigned to the pager, and is used to indicate the source of the message so as to enable the pager proxy to retrieve the appropriate public decryption key (pb.sender), and for use in authentication of the sender and for display by a receiving pager. Preferably, this number is hard-coded into memory so that it cannot easily be altered, and in current U.S. paging systems is in the form of a ten digit number.

The header data (HdrData) of the second field includes an application header (AH), which included in a field having variable length and string value, the address mode and destination header (AM/DH), the user identification number (UID), which is the same as the one included in field 1, and a message number (MSGNO) and message authentication code (MAC). In addition, e-mail address protocols require a byte indicative of address length to be added where the address mode indicates an e-mail address.

For purposes of the present invention, the message number can be any arbitrary number, although the use of a time-related reference, as allowed by the Re-FLEX protocol, is useful for account tracking or billing purposes, and in addition can be used to ensure that received message is not a recording of a message sent earlier and intercepted by an unauthorized party. For example, the message number has previously been defined as the number of seconds since January 1, 1970.

The message authentication code is a checksum used to verify that the recovered message is identical to the original message, and may be computed using an error correction code function such the cyclic recovery code (CRC) function, with CRCs being used in the illustrated embodiment or, alternatively, by computing a hash or one-way combination of the header data with the message and the session key, using an algorithm such as SHA1. By combining the message with other data to obtain the message authentication code in a way that can only recreated if the data used to recreate the code is the same as the data originally used to generate the code, the code can be used to authenticate the message, *i.e.*, to verify that the message has not been altered since the time when the code was first generated, as will be described in more detail below. It will be appreciated that the exact form of the message authentication code is not a part of the present invention, and that any message authentication code may be used so long as it can be used to authenticate the message in the manner described below.

The three blocks above the header data in Fig. 2 indicate the source of the data for the various fields. The manner in which the data is combined to form the fields is described in more detail in connection with Figs. 3-10, but the sources of the data may be summarized as (i) information entered by the user, which consists of the message (MSG) and the recipient address which forms the destination header, (ii) information stored in memory, including private and public keys of the pager, a public key of the pager proxy server, an access code which is to be compared with an access code input by the user, the encryption method indicator (EM), the user identification number (UID), and the application header, and (iii) information generated at runtime, *i.e.*, during assembly of the packet header, including the session key (SESKey), the message number (MSGNO), the addressing mode (AM), and the message authentication code (MAC).

The details of the manner in which the data shown in Fig. 2 is assembled by sending pager 1 to form the header shown in Fig. 2 is illustrated in the functional block diagram of Fig. 3, as well as the flowchart of Fig. 7. As illustrated in Fig. 3, the pager 1 includes a user input 20 connected to keys 4 or stylus 5, which supplies the destination header (DH) to a functional block 21 which assembles the header data (HdrData), and to a functional block 22 which computes the message authentication code (MAC). In addition, the user input 20 supplies the message to functional block 28, the output of which is field 3 of the header.

Pager 1 also includes a memory 24 which stores the encryption method (EM), the application header (AH), the user identification number (UID) and the encryption method identifier (EM), which are supplied directly to functional block 23 for inclusion in field 1, the user identification number and application header being also supplied to functional block 21 for inclusion in the header data, which in turn is supplied to functional block 22 for inclusion in the message authentication code. The address mode (AM), which is associated with the destination header (DH) in the header data is generated by an address mode generator 25 which can be in the form of a look-up table, device that reads a particular identifying bit in the destination header, or other device, and the message number can be generated by a counter, timer, or other device 26 depending on the nature of the message number. Finally, the session key (SESKey1) for this embodiment of the invention is an eight character string generated by a random or pseudorandom number generator 27, which supplies the session key to functional block 28 for use in encrypting the message (MSG), to functional block 22 for inclusion in the message authentication code, and to functional block 29 for encryption together with the header data by the private key of the sender. The output of functional block 29 is supplied to functional block 30 for encryption by the public key of the server, the output of block 30 serving as field 2 of the header for the packet transmitted by the sending pager.

It will be appreciated by those skilled in the art that any of the functional blocks and data or number generators illustrated in Fig. 3, or in Figs 4-6, may be implemented either by hardware or software, and that while distinguishable by function, the functions may be carried out using common subroutines, hardware, or software.

Turning to Fig. 4, the pager proxy 7 includes a database of public keys corresponding to the unique public keys of pagers registered with the encryption service provider that operates the proxy server. The database is accessed by functional block 31 according to the clear text user identification number (UID) present in the header of a packet forwarded to the pager proxy by the network operations center. Field 2 is decrypted by functional block 32 using the private key of the server (pv.server) and by functional block 33 using the public key of the sender (pb.sender) to recover the session key, and the user identification number (UID) recovered from field 2 is compared by functional block 34 with the user identification number of field 1 to verify the authenticity of field 2 and recover the session key (SESKey1). A functional block 35 then uses the session key to decrypt the message (MSG).

The message recovered by the pager proxy is authenticated in functional block 37, by comparing the message authentication code recovered from field 2 with the output of a functional block 36 that computes the message authentication code based on the destination header (DH), application header (AH), user identification number (UID), message number (MSGNO), and session key (SESKey1) recovered from field 2, and the message recovered from field 3. The message, session key, and header data (HdrData) are then made available by functional block 38 to an encryption or repackaging module, illustrated in Fig. 5, for repackaging in a way that will enable decryption by a destination pager.

As shown in Fig. 5, the application header (AH) and message number (MSGNO) received from functional block 38 is provided to functional blocks 41 and 42 for inclusion in the header data and message authentication code, while the address mode (AM) and encryption method (EM) obtained from field 1 of the packet received from the sender is passed to functional block 43 or regenerated for inclusion as clear text in the packet header. In order to permit decryption and authentication of the repackaged header by the receiving pager, however, the destination header (DH) and user identification number (UID) are swapped, so that the original destination header is supplied by the pager proxy to functional blocks 41, 42, and 43 as the user identification number (UID), and the original user identification number are supplied to functional blocks 41 and 42 as the destination header (DH). Functional block 42 generates a message authentication code based on the new destination header (DH), application header (AH), user identification number (UID), message number (MSGNO), while a new session key (SESKey2) is generated by functional block 44 in the same manner as functional block 27 shown in Fig. 3, and the resulting message authentication code (MAC) together with the new session key and header data from functional block 41 are encrypted by functional block 45 using the private key of the server (pv.server) before being sealed by functional block 46 using the public key of the destination pager (pb.recipient) and included in the header as field 2. Functional block 47 receives the message and new session key and re-encrypts the message using the new session key and an algorithm such as RC4 to generate field 3, fields 1-3 being assembled into a packet 50 for transmission to the destination pager 2 via the network operations center 3.

Again, those skilled in the art will appreciate that all of the functional blocks illustrated as being present in the proxy server and/or proxy authentication module may be implemented as software, hardware, or a combination of software and hardware, and may be varied depending on the encryption method and requirements of the pager protocol.

In addition, those skilled in the art will appreciate that the illustrated embodiment could be modified by eliminating the session key and instead using public key encryption of the message. Alternatively, instead of having the pager proxy perform any decryption of the message, the original session key could simply be re-encrypted by the pager proxy using at least the public key of the destination pager as described above, or a secret key shared with the destination pager, in which the encrypted message would simply be forwarded to the destination pager unit with the session key re-encrypted so that it can be recovered by the destination pager. While neither of these options is currently preferred because elimination of the session key leaves transmissions vulnerable to recording, and elimination of message decryption by the pager proxy makes message authentication more difficult, they should nevertheless be considered to be within the scope of the invention.

Turning to Fig. 6, the destination pager 2 includes functional blocks mirroring those of the server for decrypting messages and authenticating packets received from the pager proxy 7 via the network operations center 3. These include functional block 51 for retrieving the server public key (pb.server) from memory, functional blocks 52 and 53 for decrypting the field 2 using the recipient private key (pv.recipient) and the server public key, functional block 54 for comparing the user identification number recovered from field 2 with the user identification number in field 1, functional block 56 for decrypting the message (MSG) using the session key (SESKey2) recovered from field 2, and functional blocks 57 and 58 for generating a message authentication code and comparing it with the message authentication code recovered from field 2. It will be noted that functional block 57 may also be used to generate a message authentication code for an outgoing message, avoiding duplication of the hardware or software which performs this function.

Finally, destination pager 2 includes a functional block 59 for displaying the message (MSG) and destination header (DH) corresponding to the user identification number of the sending pager, and for alerting the user as necessary that a message has been received. The display is identical to that used for an unencrypted message, and thus the decryption operation is entirely transparent to the user.

The method steps that implement the functions illustrated in Figs. 3-6 are as follows:

First, as shown in Fig. 7, upon input of a message and destination address by the user of a pager (step 100), which may follow the input and verification of a password (not shown), a message number, address mode, and session key are generated (step 110) and the encryption method identifier, application header, user identification number, server public key, and sender private key are retrieved from memory (step 120). The encryption method identifier, address mode, and user identification number are included in field 1 (step 130), a message authentication code based on the destination header, application header, user identification number, message number, message, and session key is computed (step 140), and the application header, user identification number, destination header, message number, message authentication code, and session key are encrypted by the private key of the sending pager (step 150) and then by the public key of the pager proxy (step 160) to obtain field 2 of the packet header. Finally, the message is encrypted by the session key (step 170) to obtain field 3, and the packet header is transmitted via the network operations center to the pager proxy (step 180).

Upon receipt by the pager proxy, as shown in Fig. 8, the public key of the sending pager is retrieved based on the user identification number in field 1 (step 200), and field 2 of the packet is decrypted by the private key of the server (step 210) and then by the public key of the sending pager (step 220) based on the encryption method identified by the identifier in field 1. Authentication of the sender is provided by comparing the user identification number recovered from field 2 with the user identification number in field 1 (step 230), the message included in field 3 is decrypted using the session key recovered from field 2 (step 240), and authentication of the message is provided by generating a message authentication code based on the destination header, application header, user identification number, message number, and session key recovered from field 2 together with the decrypted message (step 250), and by then comparing the computed message authentication code with the message authentication code recovered from field 2 (step 260).

As illustrated in Fig. 9, after authenticating the information contained in field 2, the proxy server generates a new session key (step 300), encrypts the message using the new session key (step 310), assigns the original user identification as the new destination header and the original destination header as the new user identification number, computes a new message authentication code (step 330), encrypts the address header, message number, new user identification number, new destination header, new session key, and new message authentication code using the private key of the server (step 340), encrypts the result of step 340 using the public key of the destination pager (step 350), and assembles the header and packet for RF transmission to the destination pager via the network operations center (step 360).

As illustrated in Fig. 10, upon receipt by the destination pager, as shown in Fig. 8, the public key of the pager proxy server is retrieved based on the user identification number in field 1 (step 400), and field 2 of the packet is decrypted by the private key of the destination pager (step 410) and then by the public key of the pager proxy server (step 420) based on the encryption method identified by the identifier in field 1. Authentication of the sender is provided by comparing the user identification number recovered from field 2 with the user identification number in field 1 (step 430), the message included in field 3 is decrypted using the session key recovered from field 2 (step 440), and authentication of the message is provided by computing a message authentication code based on the destination header, application header, user identification number, message number, and session key recovered from field 2 together with the decrypted message (step 450), and by then comparing the computed message authentication code with the message authentication code recovered from field 2 (step 460). Finally, after authentication of the user identification number and message, the user is alerted that a message has been received and the decrypted message and information contained in the destination header are displayed at the request of the user (step 470).

Having thus described a preferred embodiment of the invention in sufficient detail to enable those skilled in the art to practice the invention, it is nevertheless anticipated that numerous variations and modifications of the invention will occur to those skilled in the art, and it is intended that all such variations and modifications be included within the scope of the invention. For example, although the preferred embodiment of the invention has the pager proxy re-package the message by first decrypting it, and then re-encrypting it using a new session key, it is also within the scope of the invention to have the pager proxy decrypt only the session key and re-encrypt the same session key using the public key or shared secret key of the destination pager. Accordingly, it is intended that the above description not be taken as limiting, but rather that it be defined solely by the appended claims.

I claim:

1. A system for adding encryption services to an existing pager network, the pager network including a network operations center which provides a means for receiving an alphanumeric message from any of a plurality of handheld pager units and forwarding the alphanumeric message to another of the plurality of handheld pager units, at least two of said pager units comprising:

means for inputting an alphanumeric message and a destination address;

means for including the alphanumeric message in a packet for transmission to the destination address by wireless transmission via the network operations center;

means for receiving an alphanumeric message from the network operations center; and

means for displaying the alphanumeric message received from the network operations center,

wherein the system for adding encryption services comprises:

means in at least one of said pager units for encrypting a message and transmitting the encrypted message via the network operations center to another of said pager units;

means in said another one of said pager units for decrypting and displaying the encrypted message; and

a pager proxy server including means for receiving a packet containing the encrypted message that has been sent to the network operations center, decrypting at least a portion of the packet, and re-encrypting said portion of the packet for delivery to said another of said pager units via said network operations center.

2. A system as claimed in claim 1, wherein said means for encrypting the message comprises means for encrypting the message by a secret key.

3. A system as claimed in claim 2, wherein said secret key is a first session key generated by a sending pager unit, said sending pager unit further comprising means for encrypting said first session key by a public key corresponding to a private key held by the pager proxy server so that the session key can be recovered only by the paging proxy server.

4. A system as claimed in claim 3, wherein said sending pager unit further comprises means for encrypting at least the first session key by a private key of the sending pager unit, and wherein said pager proxy server includes means for retrieving a public key corresponding to the private key of the sending pager unit for use as a first level authentication of the sending pager unit.

5. A system as claimed in claim 4, further comprising means for appending a unique user identification number of the sending pager unit to the header in clear text form, said user identification number being hard-coded into the sending pager unit.

6. A system as claimed in claim 5, wherein said means for encrypting at least the session key by a private key of the sending pager unit also encrypts the user identification number of the sending pager unit, and said paging proxy server includes means for decrypting the encrypted user identification number together with the first session key and comparing it with the clear text user identification number in order to authenticate the contents of the field containing the encrypted user identification number and first session key.

7. A system as claimed in claim 4, wherein the sending pager unit further comprises means for generating a first message authentication code based on various header data and the message and encrypting the various information together with the session key and the first message authentication code using the private key of the sending pager unit, and wherein the pager proxy server further comprises means for decrypting the various header

data, first message authentication code, and session key using a public key corresponding to the private key of the sending pager unit, decrypting the message using the session key, generating a second message authentication code based on the message and various header data, and comparing the first message authentication code with the second message authentication code in order to authenticate the message.

8. A system as claimed in claim 7, wherein said message authentication code is an error correction code function.

9. A system as claimed in claim 7, wherein said various header data includes at least a user identification number of the sending pager and a destination header corresponding to the input address of the destination pager.

10. A system as claimed in claim 9, wherein said various header data further includes a message number and application header.

11. A system as claimed in claim 4, wherein the sending pager further comprises means for adding an encryption method identifier in clear text to the packet header.

12. A system as claimed in claim 4, wherein an encryption algorithm used to encrypt the first session key is a public-private key encryption algorithm.

13. A system as claimed in claim 4, wherein said secret key is a first session key generated by a sending pager unit and said first session key is encrypted by a stream cipher that uses a shared secret key.

14. A system as claimed in claim 2, wherein said sending pager unit further comprises means for generating an address mode and appending the address mode in clear text to the packet header.

15. A system as claimed in claim 14, wherein said address mode indicates an address type selected from the group consisting of pager address types and e-mail address types, and wherein the pager proxy server is connected to a computer network gateway server and includes means for re-packaging said message in an e-mail packet and transmitting the e-mail packet via said computer network server to an e-mail address.

16. A system as claimed in claim 15, further comprising means for receiving e-mail packets from said computer network gateway server, and re-packaging said e-mail packets for transmission to the destination pager unit via said network operation center, and means for repackaging packets received from the network operations center for forwarding to an e-mail server.

17. A system as claimed in claim 1, wherein said means included in the pager proxy server for decrypting at least a portion of the packet includes means for decrypting, using a secret key, a portion of the packet containing a first session key used by a sending pager unit to encrypt said portion of the packet.

18. A system as claimed in claim 17, wherein said pager proxy server further includes means for decrypting said message using said first session key, means for generating a second session key, and means for re-encrypting the message using the second session key.

19. A system as claimed in claim 18, wherein said means for re-encrypting said portion of the packet includes means for encrypting the second session key by a secret key.

20. A system as claimed in claim 19, wherein said means for encrypting said portion of the packet by a secret key includes means for re-encrypting the second session key by a public key corresponding to a private key of a destination pager unit.

21. A system as claimed in claim 20, wherein said means for encrypting said portion of the packet by a secret key further includes means for, before re-encrypting the second session key by the public key corresponding to a private key of the destination pager, encrypting the second session key and various additional data by a private key of the pager proxy server.

22. A system as claimed in claim 21, wherein said additional data includes a second user identification number, said second user identification number corresponding to a first destination header included in said decrypted portion of the packet received from the sending pager unit, and wherein said destination paging unit includes means for comparing said second user identification number encrypted with said second session key to a clear text version of the second user identification number received from the pager proxy server in order to authenticate the pager proxy server.

23. A system as claimed in claim 22, wherein said additional data includes a second destination header corresponding to the first user identification number, and wherein said second pager unit includes means for displaying information included in said second destination header in order to indicate an address of the sending pager unit.

24. A system as claimed in claim 22, wherein said additional data includes a second destination header corresponding to the first user identification number, a message number recovered from said decrypted portion of the packet received from the sending pager unit, and an application number.

25. A system as claimed in claim 22, wherein said pager proxy server further comprises means for generating a message authentication code based on said message, said second session key, and said additional data, and said destination pager unit includes means for recovering said additional data and computing a message authentication code based on the additional data, said second session key, and said message in order to authenticate said message.

26. An encryption method according to which encryption services may be added to an existing two-way wireless pager network, the pager network including a network operations center which provides a means for receiving an alphanumeric message from any of a plurality of handheld pager units and forwarding the alphanumeric message to another of the plurality of handheld pager units, comprising the steps of:

causing one of said pager units to perform the steps of encrypting a message, including the encrypted message in a wireless transmission packet, and transmitting the encrypted message from said one of said pager units to a pager proxy server via the network operations center;

causing the pager proxy server to perform the steps of receiving the encrypted message and repackaging it for transmission to another of said pager units via the network operations center; and

causing said another of said pager units to perform the steps of decrypting and displaying the encrypted message.

27. A method as claimed in claim 26, wherein the step of encrypting the message comprises the step of encrypting the message by a secret key corresponding to a secret key of the pager proxy server so that the session key can only be recovered by the paging proxy server.

28. A method as claimed in claim 26, wherein said secret key is a first session key generated by a sending pager unit, and wherein said sending pager unit further performs the step of encrypting said first session key by a public key corresponding to a private key held by the pager proxy server.

29. A method as claimed in claim 27, wherein said sending pager unit further performs the step of encrypting at least the first session key by a private key of the sending pager unit, and wherein said pager proxy server performs the step of retrieving a public key corresponding to the private key of the sending pager unit for use as a first

level authentication of the sending pager unit.

30. A method as claimed in claim 29, further comprising of the step of appending a unique user identification number of the sending pager unit to the header of the transmission to the paging proxy server in clear text form, said user identification number being hard-coded into the sending pager unit.

31. A method as claimed in claim 30, wherein said step of encrypting at least the session key by a private key of the sending pager unit includes the step of encrypting the user identification number of the sending pager unit, and said paging proxy server further performs the steps of decrypting the encrypted user identification number together with the first session key and comparing it with the clear text user identification number in order to authenticate the contents of the field containing the encrypted user identification number and first session key.

32. A method as claimed in claim 29, wherein the sending pager unit further performs the step of computing a first message authentication code based on various header data and the message and encrypting the various information together with the session key and the first message authentication code using the private key of the sending pager unit, and wherein the pager proxy server further performs the steps of decrypting the various header data, first message authentication code, and session key using a public key corresponding to the private key of the sending pager unit, decrypting the message using the session key, generating a second message authentication code based on the message and various header data, and comparing the first message authentication code with the second message authentication code in order to authenticate the message.

33. A method as claimed in claim 32, wherein said message authentication code is an error correction code function.

34. A method as claimed in claim 32, wherein said various header data includes at least the user identification number of the sending pager and a destination header corresponding to the input address of the destination pager.

35. A method as claimed in claim 34, wherein said various header data further includes a message number and application header.

36. A method as claimed in claim 34, wherein the sending pager further performs the step of adding an encryption method identifier in clear text to the packet header.

37. A method as claimed in claim 29, wherein an encryption algorithm used to encrypt the first session key is a public-private key encryption algorithm.

38. A method as claimed in claim 27, wherein said secret key is a first session key generated by a sending pager unit and said first session key is encrypted by a stream cipher that uses a shared secret key.

39. A method as claimed in claim 37, wherein said sending pager unit further performs the step of generating an address mode and appending the address mode in clear text to the packet header.

40. A method as claimed in claim 39, wherein said address mode indicates an address type selected from the group consisting of pager address types and e-mail address types, and wherein the pager proxy server is connected to a computer network gateway server and further performs the step of re-packaging said message in an e-mail packet and transmitting the e-mail packet via said computer network server to an e-mail address.

41. A method as claimed in claim 40, further performs the steps of receiving e-mail packets from said computer network gateway server, and re-packaging said e-mail

packets for transmission to the destination pager unit via said network operation center.

42. A method as claimed in claim 26, wherein said step of repackaging the encrypted message for transmission includes the step of causing the pager proxy server to encrypt, using a secret key, a portion of the packet containing a first session key used by a sending pager unit to encrypt said portion of the packet.

43. A method as claimed in claim 42, wherein said pager proxy server further performs the steps of decrypting said message using said first session key, generating a second session key, and re-encrypting the message using the second session key.

44. A method as claimed in claim 43, wherein said pager proxy server further performs the step of encrypting the second session key by a secret key.

45. A method as claimed in claim 44, wherein said step of encrypting said portion of the packet by a secret key includes the step of re-encrypting the second session key by a public key corresponding to a private key of a destination pager unit.

46. A method as claimed in claim 45, wherein said step of encrypting said portion of the packet by a secret key further includes the step of, before re-encrypting the second session key by the public key corresponding to a private key of the destination pager, encrypting the second session key and various additional data by a private key of the pager proxy server.

47. A method as claimed in claim 46, wherein said additional data includes a second user identification number, said second user identification number corresponding to a first destination header included in said decrypted portion of the packet received from the sending pager unit, and wherein said destination paging unit perform the step of comparing said second user identification number encrypted with said second session key

to a clear text version of the second user identification number received from the pager proxy server in order to authenticate the pager proxy server.

48. A method as claimed in claim 47, wherein said additional data includes a second destination header corresponding to the first user identification number, and wherein said second pager unit performs the step of displaying information included in said second destination header in order to indicate an address of the sending pager unit.

49. A method as claimed in claim 47, wherein said additional data includes a second destination header corresponding to the first user identification number, a message number recovered from said decrypted portion of the packet received from the sending pager unit, and an application number.

50. A method as claimed in claim 47, wherein said pager proxy server further performs the step of computing a message authentication code based on said message, said second session key, and said additional data, and said destination pager unit further performs the step of recovering said additional data and computing a message authentication code based on the additional data, said second session key, and said message in order to authenticate said message.

51. A two-way alphanumeric pager unit, comprising:

means for inputting a message and a destination address;

means for generating a session key;

means for encrypting the message using the session key;

means for protecting the session key so that it can only be recovered by a pager proxy server;

means for transmitting the message via a wireless pager network to the pager proxy server;

means for receiving an encrypted message transmitted via the wireless pager network from the pager proxy server;

means for decrypting an encrypted session key appended to the message;

means for decrypting the encrypted message transmitted from the pager proxy server using the decrypted session key; and

means for displaying the message.

52. A pager unit as claimed in claim 51, wherein said means for protecting the session key comprises means for encrypting the session key by a secret key.

53. A pager unit as claimed in claim 52, wherein said secret key is a first session key generated by the pager unit, said sending pager unit further comprising means for encrypting said first session key by a public key corresponding to a private key held by the pager proxy server.

54. A pager unit as claimed in claim 53, further comprising means for appending a unique user identification number of the pager unit to the header in clear text form, said user identification number being hard-coded into the pager unit.

55. A pager unit as claimed in claim 54, wherein said means for encrypting at least the session key by a secret key also encrypts the user identification number of the sending pager unit, said encrypted user identification number being compared by the pager proxy server with a clear text version of the user identification number transmitted with a packet header in order to authenticate the pager unit.

56. A pager unit as claimed in claim 55, wherein the pager unit further comprises means for computing a message authentication code based on various header data and the message, and means for encrypting the various information together with the session key and the message authentication code using a private key of the sending pager unit in order to provide a means for authentication by the pager proxy of the message.

57. A pager unit as claimed in claim 56, wherein said message authentication code is an error correction code function.

58. A pager unit as claimed in claim 57, wherein said various header data includes at least the user identification number of the pager unit and a destination header corresponding to the input address of a destination pager.

59. A pager unit as claimed in claim 58, wherein said various header data further includes a message number and application header.

60. A pager unit as claimed in claim 52, wherein the pager unit further comprises means for adding an encryption method identifier in clear text to a packet header.

61. A pager unit as claimed in claim 60, wherein an encryption algorithm used to encrypt the first session key is a public-private key encryption algorithm.

62. A pager unit as claimed in claim 60, wherein said secret key is a first session key generated by a sending pager unit and said first session key is encrypted by a stream cipher that uses a shared secret key.

63. A pager unit as claimed in claim 62, wherein said pager unit further comprises means for generating an address mode and appending the address mode in clear text to the packet header.

64. A pager unit as claimed in claim 62, wherein said address mode is selected from the group consisting of pager address types and e-mail address types, and wherein the pager proxy server is connected to a computer network server and includes means for re-packaging said message in an e-mail packet and transmitting the e-mail packet via said computer network server to an e-mail address.

65. A pager proxy server, comprising:

means for receiving a message encrypted by a session key, the session key being encrypted and appended to the encrypted message, from a network operations center of a pager network;

means for recovering the session key using a secret key of the server;

means for authenticating the sender of the message; and

means for re-transmitting the message encrypted by a session key in a manner which enables decryption of the message only by a holder of a second secret key.

66. A server as claimed in claim 65, wherein said means for re-transmitting the message comprises means for decrypting the message using the first session key, re-encrypting the message using a second session key, and encrypting the second session key.

67. A server as claimed in claim 66, wherein said first secret key is a private key held by the pager proxy server.

68. A server as claimed in claim 67, further comprising means for retrieving a public key corresponding to a private key of a sending pager unit for use as a first level authentication of the sending pager unit.

69. A server as claimed in claim 68, further comprising means for decrypting the a user identification number of the sending pager unit together with the session key and comparing it with a clear text user identification number in order to authenticate the contents of the field containing the encrypted user identification number and session key.

70. A server as claimed in claim 69, further comprising means for decrypting various header data, a first message authentication code, and a session key using a public key corresponding to the private key of the sending pager unit, decrypting the message using

the session key, generating a second message authentication code based on the message and various header data, and comparing the first message authentication code with the second message authentication code in order to authenticate the message.

71. A server as claimed in claim 70, wherein said message authentication code is an error correction code function.

72. A server as claimed in claim 70, wherein said various header data includes at least the user identification number of the sending pager and a destination header corresponding to the input address of the destination pager.

73. A server as claimed in claim 72, wherein said various header data further includes a message number and application header.

74. A server as claimed in claim 73, wherein said encryption method is a public-private key encryption algorithm.

75. A server as claimed in claim 73, wherein said encryption method is RC4 secret key encryption.

76. A server as claimed in claim 72, further comprising means for receiving e-mail packets from said computer network server, and re-packaging said e-mail packets for transmission to the destination pager unit via said network operation center.

77. A system for adding encryption services to an existing pager network, the pager network including a network operations center which provides a means for receiving an alphanumeric message from any of a plurality of handheld pager units and forwarding the alphanumeric message to another of the plurality of handheld pager units, at least one of said pager units comprising:

means for inputting an alphanumeric message and a destination address;

means for including the alphanumeric message in a packet for transmission to the destination address by wireless transmission via the network operations center;

means for receiving an alphanumeric message from the network operations center; and

means for displaying the alphanumeric message received from the network operations center,

wherein the system for adding encryption services comprises:

means in at least one of said pager units for decrypting and displaying an encrypted message; and

a pager proxy server including means for receiving a packet containing the encrypted message, decrypting at least a portion of the packet, and re-encrypting said portion of the packet for delivery to said at least one of said pager units via said network operations center.

78. An alphanumeric pager unit, comprising:

means for receiving an encrypted message transmitted via a wireless pager network from a pager proxy server;

means for decrypting an encrypted session key appended to the message;

means for decrypting the encrypted message transmitted from the pager proxy server using the decrypted session key; and

means for displaying the message.

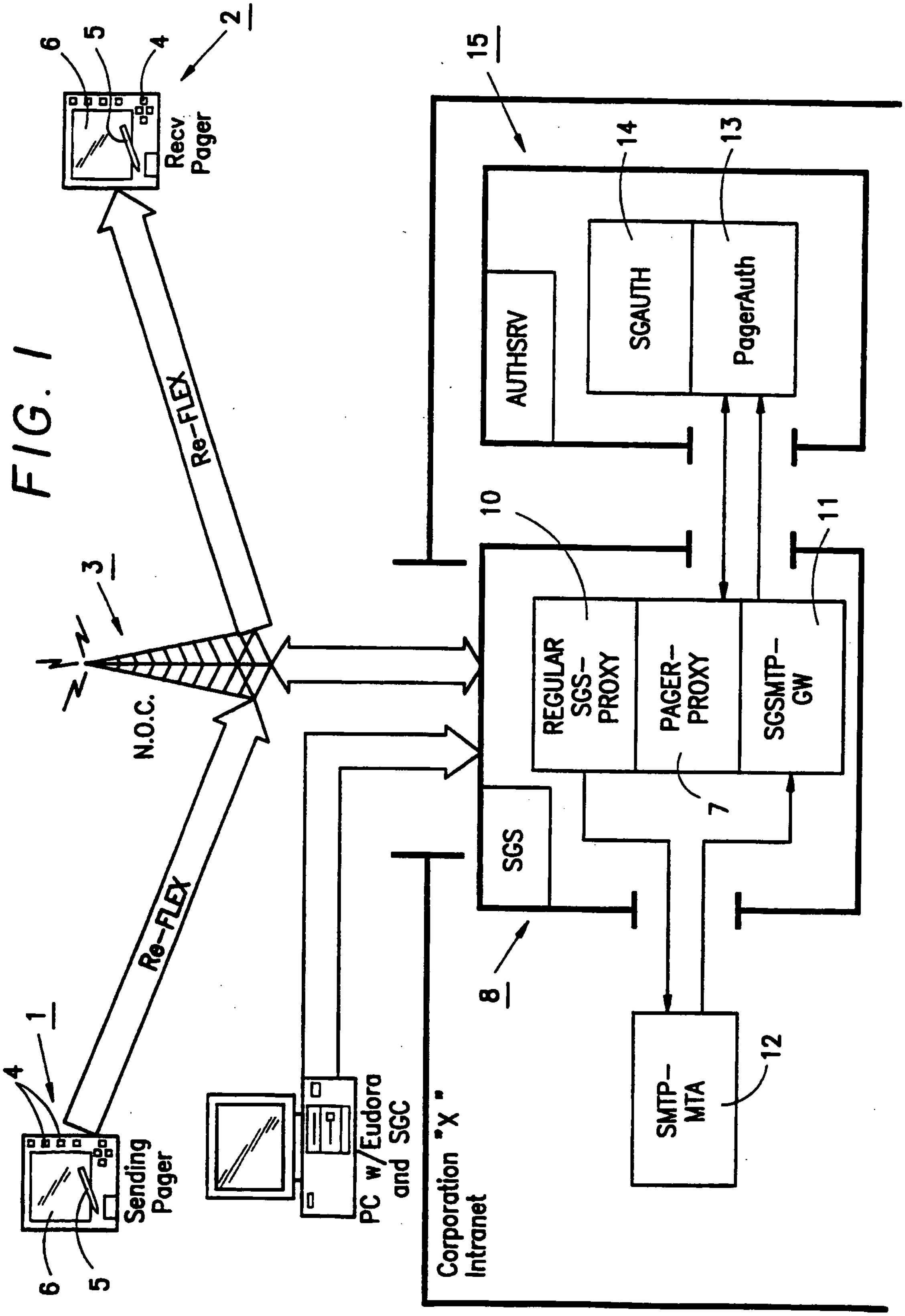

Recv Pager
Sending Pager
Re-FLEX
Re-FLEX
N.O.C.
PC w/Eudora and SGC
Corporation "X" Intranet
SGS
REGULAR SGS-PROXY
PAGER-PROXY
SGSMTP-GW
AUTHSRV
SGAUTH
PagerAuth
SMTP-MTA
1
2
3
4
5
6
7
8
10
11
12
13
14
15

FIG. 1

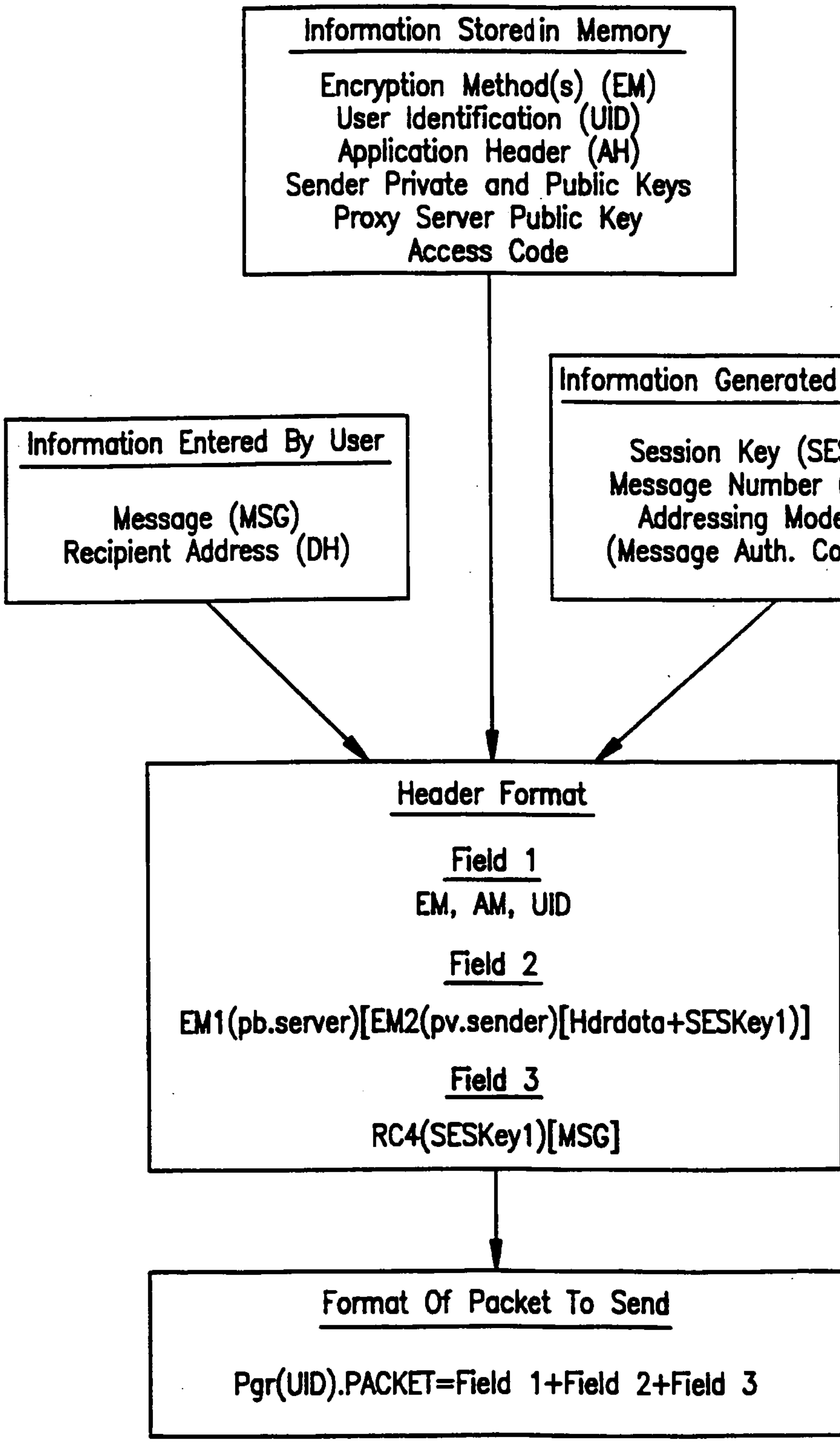
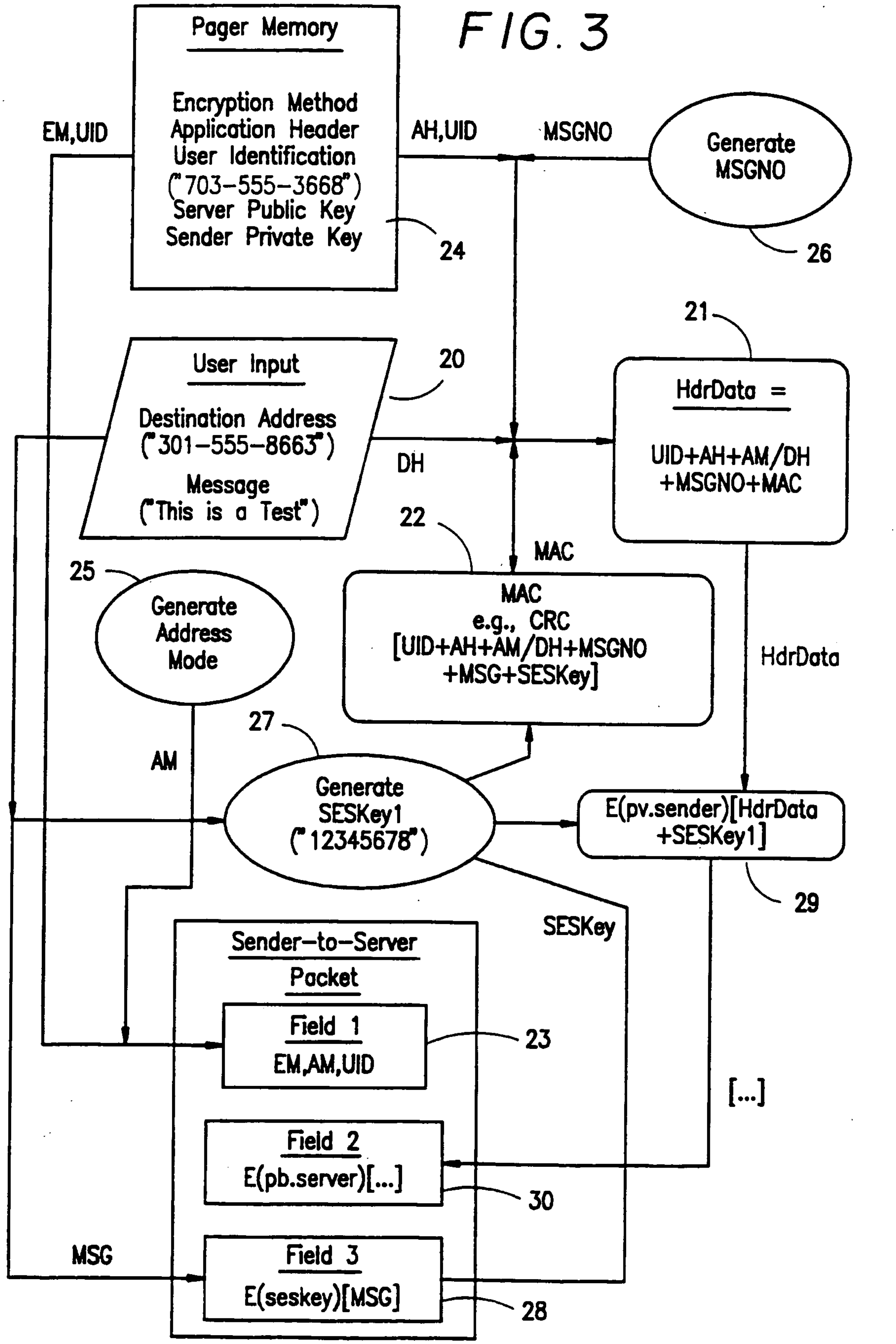

Information Stored in Memory
Encryption Method(s) (EM)
User Identification (UID)
Application Header (AH)
Sender Private and Public Keys
Proxy Server Public Key
Access Code
Information Entered By User
Message (MSG)
Recipient Address (DH)
Information Generated At Runtime
Session Key (SESKEY1)
Message Number (MSGNO)
Addressing Mode (AM)
(Message Auth. Code) MAC
Header Format
Field 1
EM, AM, UID
Field 2
EM1(pb.server)[EM2(pv.sender)[Hdrdata+SESKey1)]
Field 3
RC4(SESKey1)[MSG]
Format Of Packet To Send
Pgr(UID).PACKET=Field 1+Field 2+Field 3

FIG. 2

FIG. 3


Pager Memory
Encryption Method
Application Header
User Identification
("703-555-3668")
Server Public Key
Sender Private Key
24
EM,UID
AH,UID
MSGNO
Generate MSGNO
26
21
User Input
Destination Address
("301-555-8663")
Message
("This is a Test")
20
DH
HdrData =
UID+AH+AM/DH
+MSGNO+MAC
22
MAC
25
Generate Address Mode
MAC
e.g., CRC
[UID+AH+AM/DH+MSGNO
+MSG+SESKey]
HdrData
27
AM
Generate
SESKey1
("12345678")
E(pv.sender)[HdrData
+SESKey1]
29
SESKey
Sender-to-Server
Packet
Field 1
EM,AM,UID
23
[...]
Field 2
E(pb.server)[...]
30
MSG
Field 3
E(seskey)[MSG]
28

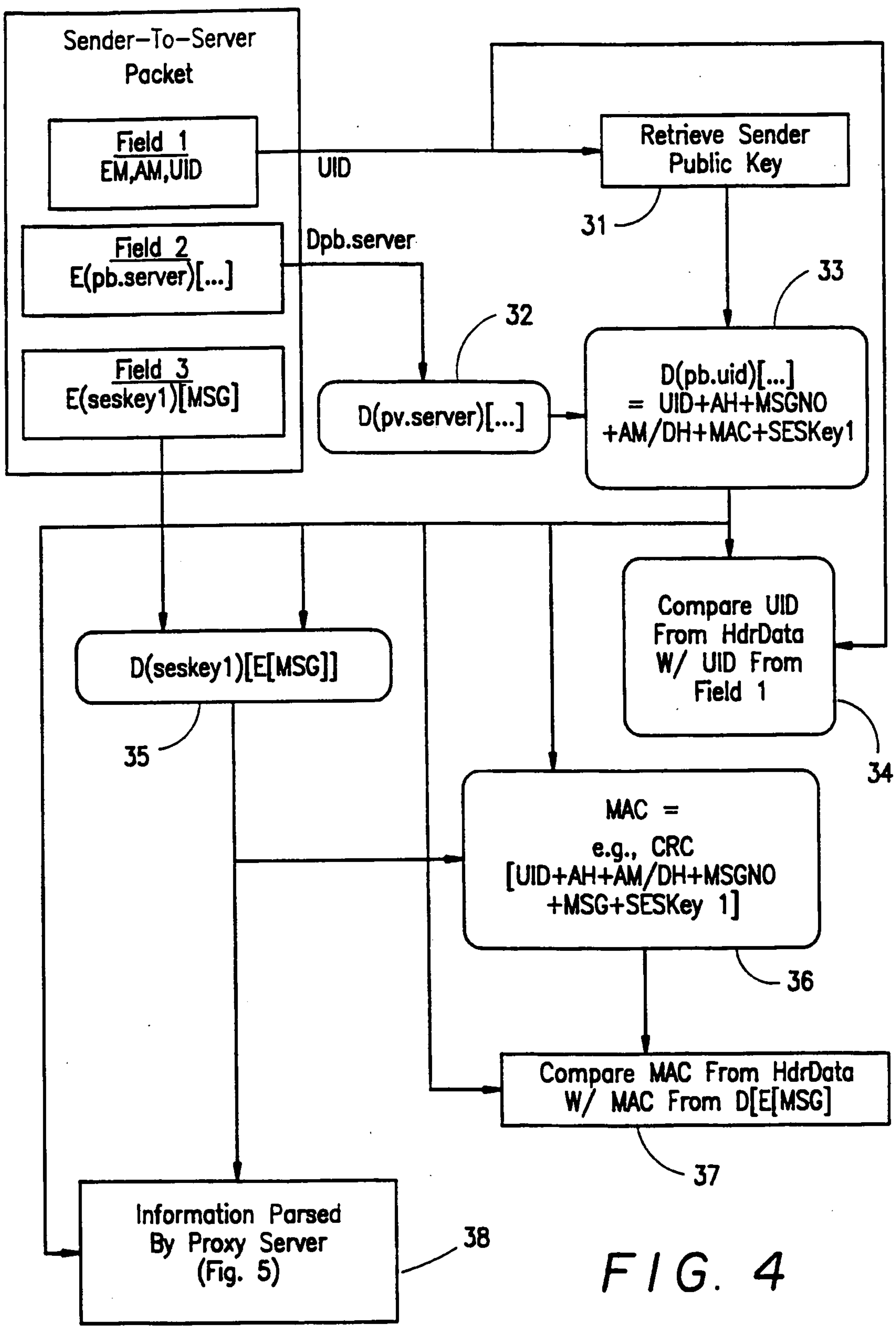

Sender–To–Server Packet
Field 1
EM,AM,UID
Field 2
E(pb.server)[...]
Field 3
E(seskey1)[MSG]
UID
Dpb.server
Retrieve Sender Public Key
31
33
32
D(pv.server)[...]
D(pb.uid)[...]
= UID+AH+MSGNO
+AM/DH+MAC+SESKey1
Compare UID From HdrData W/ UID From Field 1
34
D(seskey1)[E[MSG]]
35
MAC =
e.g., CRC
[UID+AH+AM/DH+MSGNO
+MSG+SESKey 1]
36
Compare MAC From HdrData W/ MAC From D[E[MSG]
37
Information Parsed By Proxy Server (Fig. 5)
38

FIG. 4

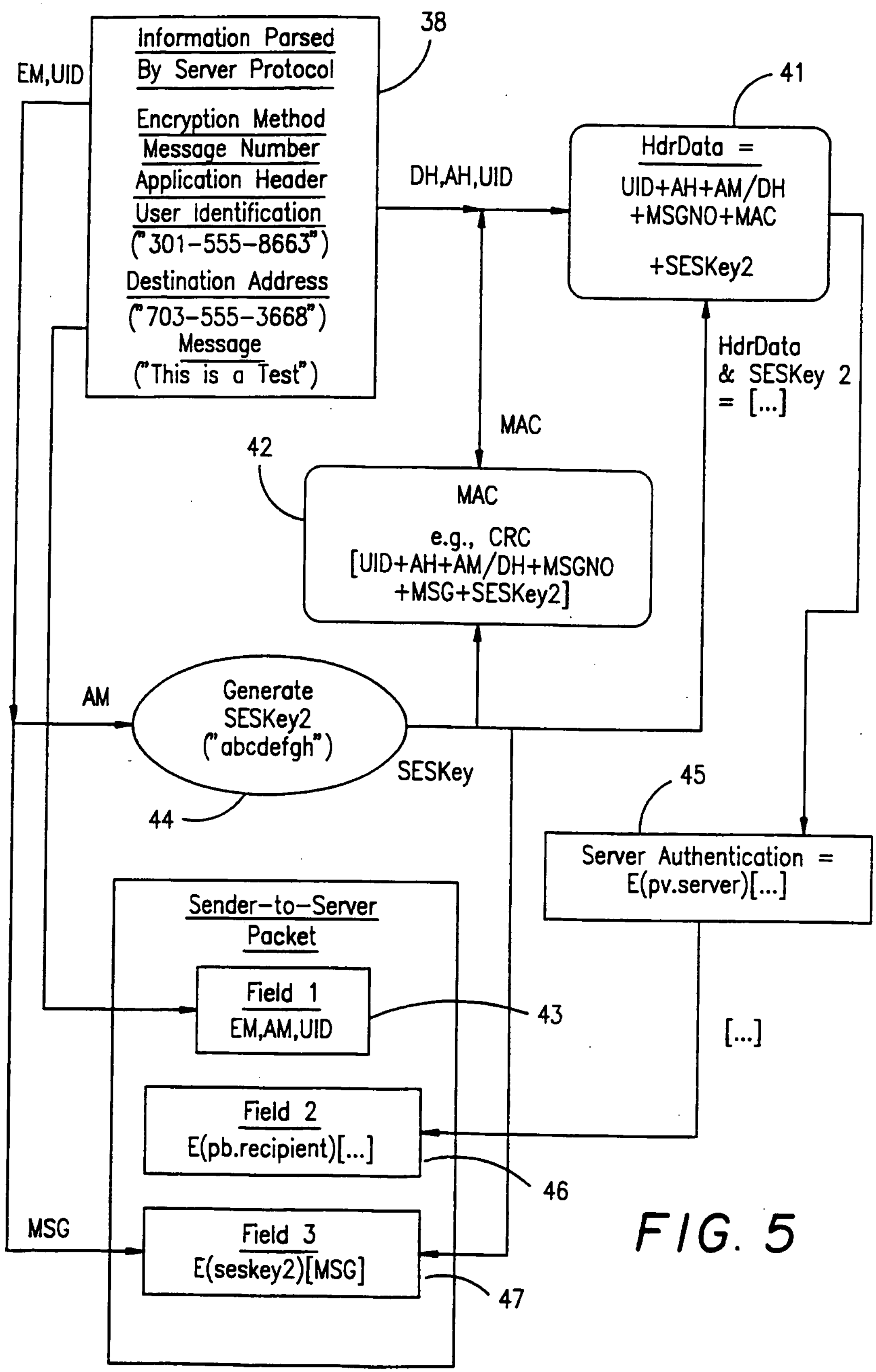

Information Parsed
By Server Protocol
Encryption Method
Message Number
Application Header
User Identification
("301-555-8663")
Destination Address
("703-555-3668")
Message
("This is a Test")
38
EM,UID
DH,AH,UID
41
HdrData =
UID+AH+AM/DH
+MSGNO+MAC
+SESKey2
HdrData
& SESKey 2
= [...]
MAC
42
MAC
e.g., CRC
[UID+AH+AM/DH+MSGNO
+MSG+SESKey2]
AM
Generate
SESKey2
("abcdefgh")
SESKey
44
45
Server Authentication =
E(pv.server)[...]
Sender-to-Server
Packet
Field 1
EM,AM,UID
43
[...]
Field 2
E(pb.recipient)[...]
46
MSG
Field 3
E(seskey2)[MSG]
47

FIG. 5

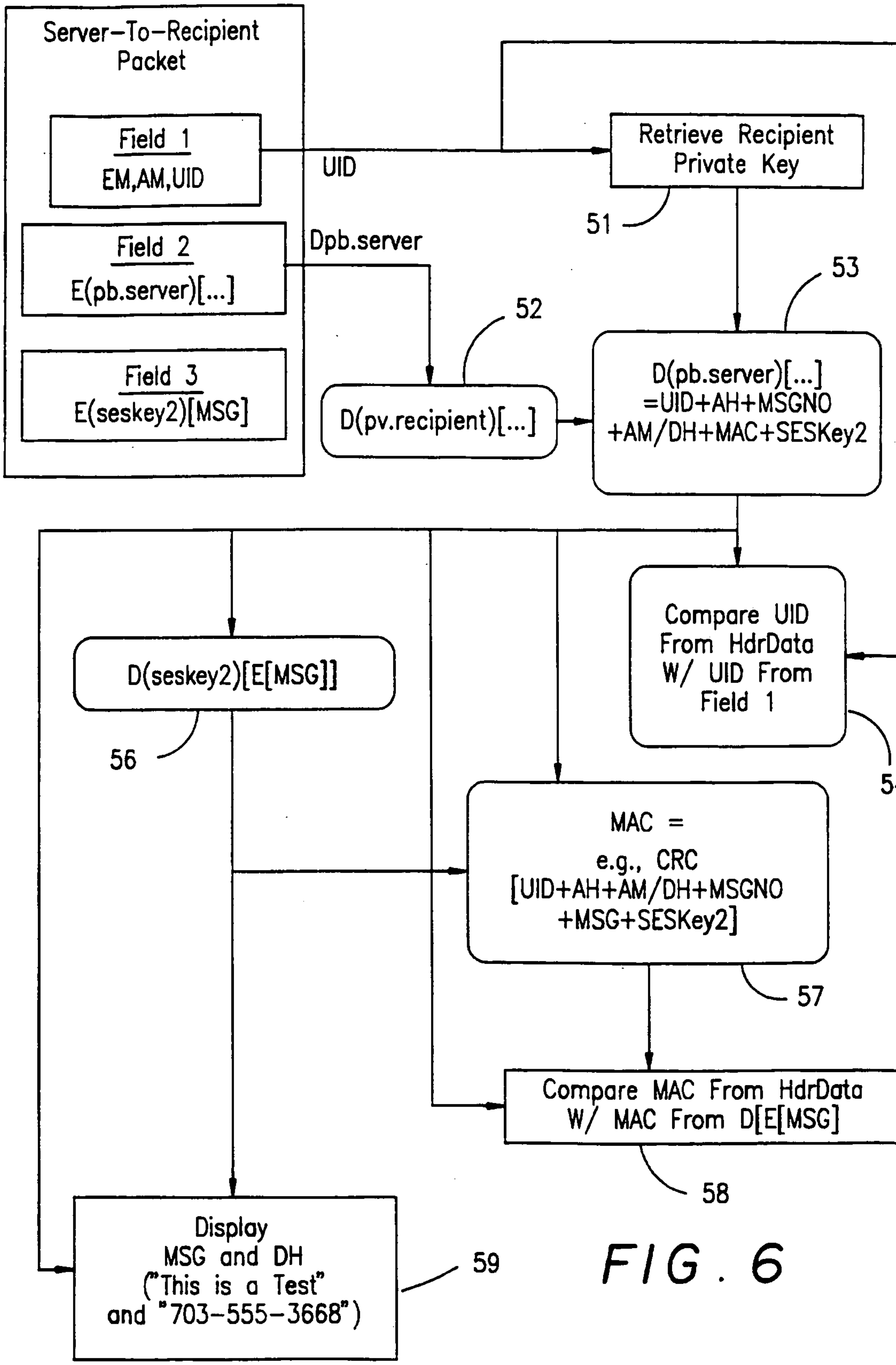

Server-To-Recipient Packet
Field 1
EM,AM,UID
Field 2
E(pb.server)[...]
Field 3
E(seskey2)[MSG]
UID
Dpb.server
Retrieve Recipient Private Key
51
52
53
D(pv.recipient)[...]
D(pb.server)[...] =UID+AH+MSGNO +AM/DH+MAC+SESKey2
Compare UID From HdrData W/ UID From Field 1
54
D(seskey2)[E[MSG]]
56
MAC = e.g., CRC [UID+AH+AM/DH+MSGNO +MSG+SESKey2]
57
Compare MAC From HdrData W/ MAC From D[E[MSG]
58
Display MSG and DH ("This is a Test" and "703-555-3668")
59

FIG. 6

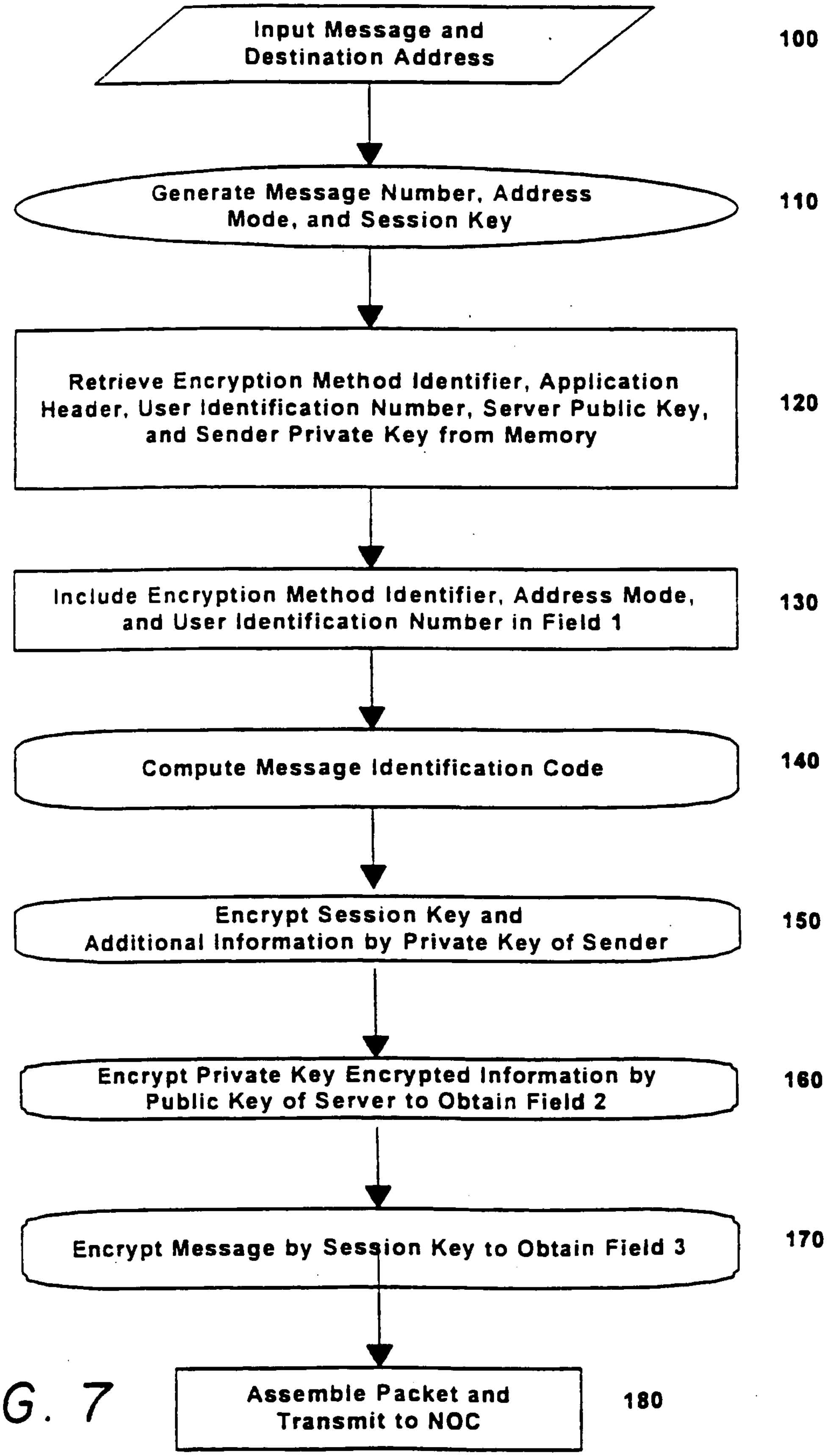
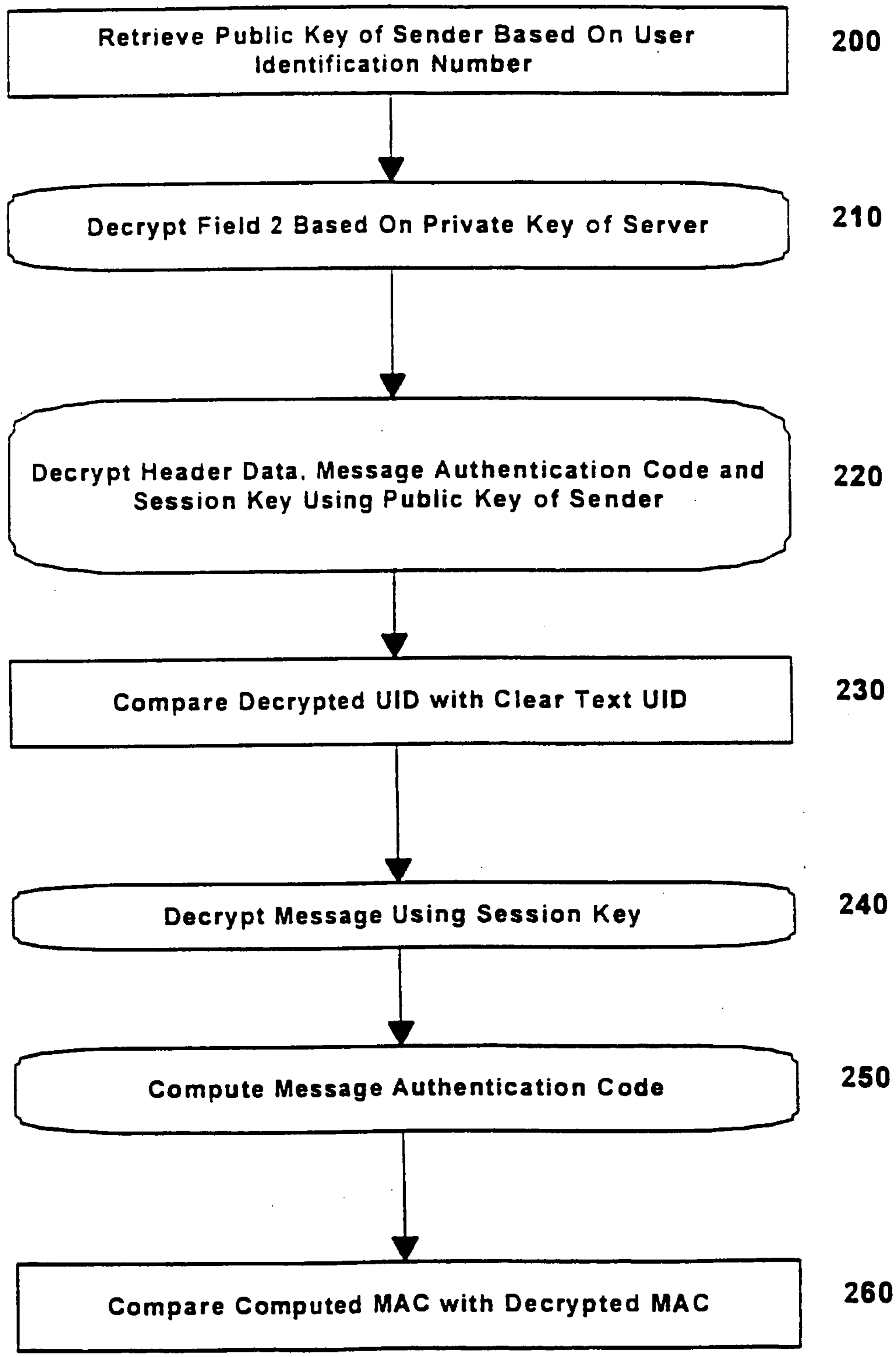

Input Message and Destination Address
100
Generate Message Number, Address Mode, and Session Key
110
Retrieve Encryption Method Identifier, Application Header, User Identification Number, Server Public Key, and Sender Private Key from Memory
120
Include Encryption Method Identifier, Address Mode, and User Identification Number in Field 1
130
Compute Message Identification Code
140
Encrypt Session Key and Additional Information by Private Key of Sender
150
Encrypt Private Key Encrypted Information by Public Key of Server to Obtain Field 2
160
Encrypt Message by Session Key to Obtain Field 3
170
Assemble Packet and Transmit to NOC
180

FIG. 7

FIG. 8


Retrieve Public Key of Sender Based On User Identification Number
200
Decrypt Field 2 Based On Private Key of Server
210
Decrypt Header Data, Message Authentication Code and Session Key Using Public Key of Sender
220
Compare Decrypted UID with Clear Text UID
230
Decrypt Message Using Session Key
240
Compute Message Authentication Code
250
Compare Computed MAC with Decrypted MAC
260

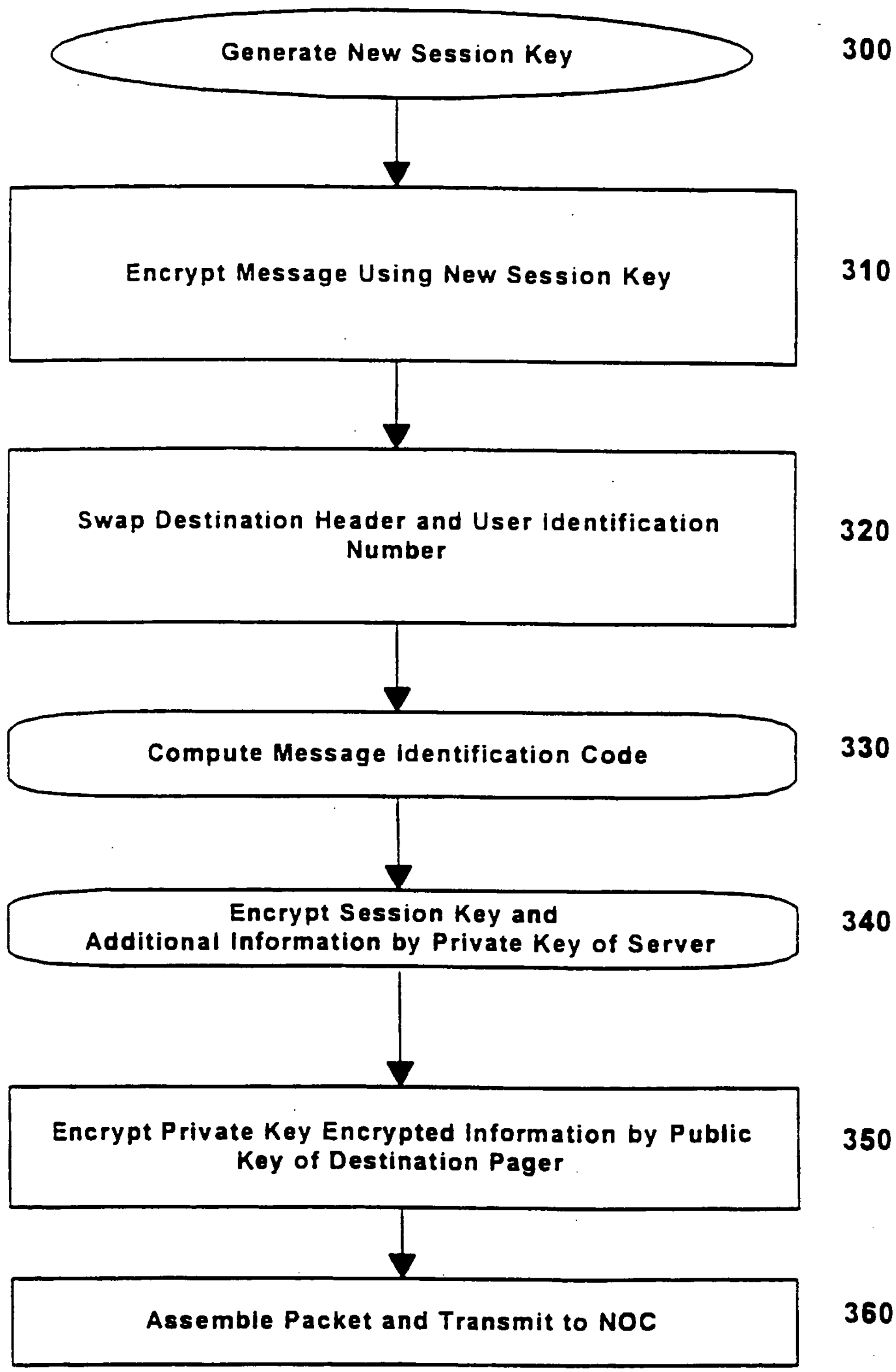
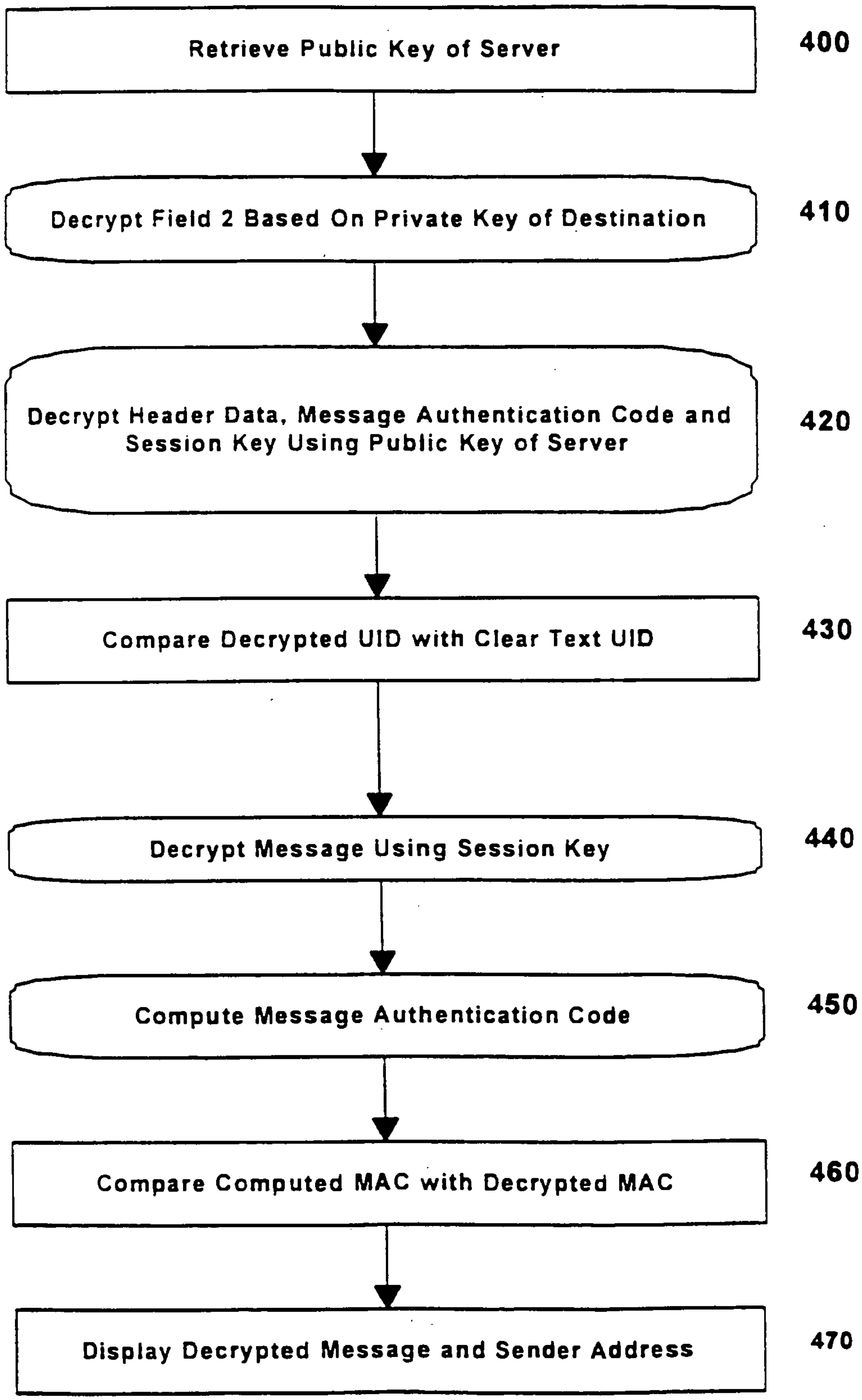

Generate New Session Key
300
Encrypt Message Using New Session Key
310
Swap Destination Header and User Identification Number
320
Compute Message Identification Code
330
Encrypt Session Key and Additional Information by Private Key of Server
340
Encrypt Private Key Encrypted Information by Public Key of Destination Pager
350
Assemble Packet and Transmit to NOC
360

FIG. 9

FIG. 10


Retrieve Public Key of Server
400
Decrypt Field 2 Based On Private Key of Destination
410
Decrypt Header Data, Message Authentication Code and Session Key Using Public Key of Server
420
Compare Decrypted UID with Clear Text UID
430
Decrypt Message Using Session Key
440
Compute Message Authentication Code
450
Compare Computed MAC with Decrypted MAC
460
Display Decrypted Message and Sender Address
470

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/27531

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04L 9/08
US CL :380/21
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. : 380/21,44,45,49

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)
Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,285,496 A (FRANK et al) 08 February 1994 (08.02.94), column 2, lines 28-44, column 4, lines 12-68, column 6, lines 11-49.	1-31,36-56, 58,60-68,74-78
Y	US 5,604,801 A (DOLAN et al) 18 February 1997 (18.02.97), abstract, column 3, lines 2-38, 50-60, column 4, lines 19-24, 40-55.	1-31,36-56 58,60-68, 74-78
A	US 5,602,918 A (CHEN et al) 11 February 1997 (11.02.97), abstract, column 2, lines 36-41,57-60, column 4, lines 43-63.	3-4,6,17-18,27-28, 31-32,40,42,53,65,68-70,77
A	US 5,452,356 A (ALBERT et al) 19 September 1995 (19.09.95), column 1, lines 60-68, column 2, lines 1-42, column 11, lines 15-55.	1-78

[X] Further documents are listed in the continuation of Box C. [ ] See patent family 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 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 17 FEBRUARY 1999	Date of mailing of the international search report 06 MAY 1999
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 GAIL HAYES Jon Hill Telephone No. (703) 305-97[illegible]1

Form PCT/ISA/210 (second sheet)(July 1992)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/27531

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,495,533 A (LINEHAN et al) 27 February 1996 (27.02.96), column 9, lines 42-58, column 10, lines 22-32.	7,9,35,59,69-70,72-73

Form PCT/ISA/210 (continuation of second sheet)(July 1992)★

B. FIELDS SEARCHED

Electronic data bases consulted (Name of data base and where practicable terms used):

APS

search terms: cypher, cipher, encode,encrypt, decrypt, key, keys, pager, wireless, proxy server, authenticate, authentication, transmission, transmitting, key management, public key, two-way communication, re-encrypt, messages, data, information

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/30 — A1

(11) International Publication Number: WO 99/35782

(43) International Publication Date: 15 July 1999 (15.07.99)

(21) International Application Number: PCT/US98/27896

(22) International Filing Date: 31 December 1998 (31.12.98)

(30) Priority Data:

60/070,344	2 January 1998 (02.01.98)	US
60/089,529	15 June 1998 (15.06.98)	US

(71) Applicant: CRYPTOGRAPHY RESEARCH, INC. [US/US]; Suite 1088, 870 Market Street, San Francisco, CA 94102 (US).

(72) Inventors: KOCHER, Paul, C.; 143 Fillmore Street, San Francisco, CA 94117 (US). JAFFE, Joshua, M.; 21B Bird Street, San Francisco, CA 94110 (US).

(74) Agents: LAURIE, Ronald, S. et al.; Skadden, Arps, Slate, Meagher & Flom LLP, 525 University Avenue, Palo Alto, CA 94301 (US).

(81) Designated States: 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, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, 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).

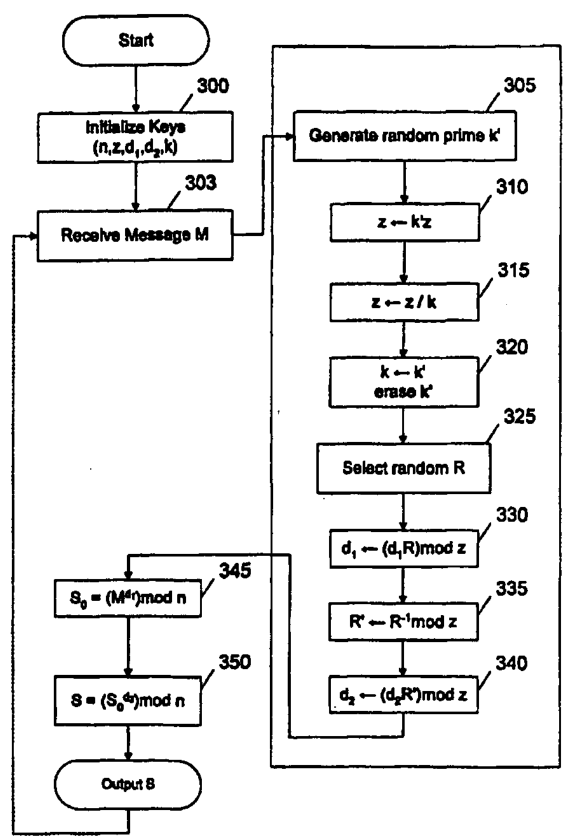
Published

With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: LEAK-RESISTANT CRYPTOGRAPHIC METHOD AND APPARATUS

(57) Abstract

The present invention provides a method and apparatus for securing cryptographic devices against attacks involving external monitoring and analysis. A "self-healing" property is introduced, enabling security to be continually re-established following partial compromises. In addition to producing useful cryptographic results, a typical leak-resistant cryptographic operation modifies or updates (330) secret key material in a manner designed to render useless any information about the secrets that may have previously leaked from the system. Exemplary leak-proof and leak-resistant implementations of the invention are shown for symmetric authentication (350), certified Diffie-Hellman (when either one or both users have certificates), RSA, ElGamal public key decryption (303).



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		

LEAK-RESISTANT CRYPTOGRAPHIC METHOD AND APPARATUS

This application claims the benefit of US Provisional Application No. 60/070,344 filed January 2, 1998, and US Provisional Application No. 60/089,529 filed June 15, 1998.

TECHNICAL FIELD

The method and apparatus of the present invention relate generally to cryptographic systems and, more specifically, to securing cryptographic tokens that must maintain the security of secret information in hostile environments.

BACKGROUND OF THE INVENTION

Most cryptosystems require secure key management. In public-key based security systems, private keys must be protected so that attackers cannot use the keys to forge digital signatures, modify data, or decrypt sensitive information. Systems employing symmetric cryptography similarly require that keys be kept secret. Well-designed cryptographic algorithms and protocols should prevent attackers who eavesdrop on communications from breaking systems. However, cryptographic algorithms and protocols traditionally require that tamper-resistant hardware or other implementation-specific measures prevent attackers from accessing or finding the keys.

If the cryptosystem designer can safely assume that the key management system is completely tamper-proof and will not reveal any information relating to the keys except via the messages and operations defined in the protocol, then previously known cryptographic techniques are often sufficient for good security. It is currently extremely difficult, however, to make hardware key management systems that provide good security, particularly in low-cost unshielded cryptographic devices for use in applications where attackers will have physical control over the device. For example, cryptographic tokens (such as smartcards used in electronic cash and copy protection schemes) must protect their keys even in potentially hostile environments. (A token is a device that contains or manipulates cryptographic keys that need to be protected from attackers. Forms in which tokens may be manufactured include, without limitation, smartcards, specialized encryption and key management devices, secure telephones, secure picture phones, secure web servers, consumer electronics devices using cryptography, secure microprocessors, and other tamper-resistant cryptographic systems.)

A variety of physical techniques for protecting cryptographic devices are known, including enclosing key management systems in physically durable enclosures, coating integrated circuits with special coatings that destroy the chip when removed, and wrapping devices with fine wires that detect tampering. However, these approaches are expensive, difficult to use in single-chip solutions (such as smartcards), and difficult to evaluate since there is no mathematical basis for their security. Physical tamper resistance techniques are also ineffective against some attacks. For example, recent work by Cryptography Research has shown that attackers can non-invasively extract secret keys using careful measurement and analysis of many devices' power consumption. Analysis of timing measurements or electromagnetic radiation can also be used to find secret keys.

Some techniques for hindering external monitoring of cryptographic secrets are known, such as using power supplies with large capacitors to mask fluctuations in power consumption, enclosing devices in well-shielded cases to prevent electromagnetic radiation, message blinding to prevent timing attacks, and buffering of inputs/outputs to prevent signals from leaking out on I/O lines. Shielding, introduction of noise, and other such countermeasures are often, however, of limited value, since skilled attackers can still find keys by amplifying signals and filtering out noise by averaging data collected from many operations. Further, in smartcards and other tamper-resistant chips, these countermeasures are often inapplicable or insufficient due to reliance on external power sources, impracticality of shielding, and other physical constraints. The use of blinding and constant-time mathematical algorithms to prevent timing attacks is also known, but does not prevent more complex attacks such as power consumption analysis (particularly if the system designer cannot perfectly predict what information will be available to an attacker, as is often the case before a device has been physically manufactured and characterized).

The present invention makes use of previously-known cryptographic primitives and operations. For example: U.S. patent 5,136,646 to Haber et al. and the pseudorandom number generator used in the RSAREF cryptographic library use repeated application of hash functions; anonymous digital cash schemes use blinding techniques; zero knowledge protocols use hash functions to mask information; and key splitting and threshold schemes store secrets in multiple parts.

SUMMARY OF THE INVENTION

The present invention introduces leak-proof and leak-resistant cryptography, mathematical approaches to tamper resistance that support many existing cryptographic primitives, are inexpensive, can be implemented on existing hardware (whether by itself or via software capable of running on such hardware), and can solve problems involving secrets leaking out of cryptographic devices. Rather than assuming that physical devices will provide perfect security, leak-proof and leak-resistant cryptographic systems may be designed to remain secure even if attackers are able to gather some information about the system and its secrets. This invention describes leak-proof and leak-resistant systems that implement symmetric authentication, Diffie-Hellman exponential key agreement, ElGamal public key encryption, ElGamal signatures, the Digital Signature Standard, RSA, and other algorithms.

One of the characteristic attributes of a typical leak-proof or leak-resistant cryptosystem is that it is "self-healing" such that the value of information leaked to an attacker decreases or vanishes with time. Leak-proof cryptosystems are able to withstand leaks of up to L_{MAX} bits of information per transaction, where L_{MAX} is a security factor chosen by the system designer to exceed to the maximum anticipated leak rate. The more general class of leak-resistant cryptosystems includes leak-proof cryptosystems, and others that can withstand leaks but are not necessarily defined to withstand any defined maximum information leakage rate. Therefore, any leak-proof system shall also be understood to be leak-resistant. The leak-resistant systems of the present invention can survive a variety of monitoring and eavesdropping attacks that would break traditional (non-leak-resistant) cryptosystems.

A typical leak-resistant cryptosystem of the present invention consists of three general parts. The initialization or key generation step produces secure keying material appropriate for the scheme. The update process cryptographically modifies the secret key material in a manner designed to render useless any information about the secrets that may have previously leaked from the system, thus providing security advantages over systems of the background art. The final process performs cryptographic operations, such as producing digital signatures or decrypting messages.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows an exemplary leak-resistant symmetric authentication method.

Figure 2 shows an exemplary leak-resistant Diffie-Hellman exponential key exchange operation.

Figure 3 shows an exemplary leak-resistant RSA private key operation.

Figure 4 shows an exemplary leak-resistant ElGamal signing operation.

DETAILED DESCRIPTION OF THE INVENTION

The sections following will describe an introduction to leak-proof/leak-resistant cryptography, followed by various embodiments of the general techniques of the invention as applied to improve the security of common cryptographic protocols.

I. Introduction and Terminology

The leakage rate L is defined as the number of bits of useful information about a cryptosystem's secrets that are revealed per operation, where an operation is a cryptographic transaction. Although an attacker may be able to collect more than L bits worth of measurement data, by definition this data yields no more than L bits of useful information about the system's secrets.

The implementer of a leak-proof system chooses a design parameter L_{MAX}, the maximum amount of leakage per operation the system may allow if it is to remain uncompromised. L_{MAX} should be chosen conservatively, and normally should significantly exceed the amount of useful information known to be leaked to attackers about the system's secrets during each transaction. Designers do not necessarily need to know accurately or completely the quantity and type of information that may leak from their systems; the choice of L_{MAX} may be made using estimates and models for the system's behavior. General factors affecting the choice of L_{MAX} include the types of monitoring potentially available to attackers, the amount of error in attackers' measurements, and engineering constraints that limit L_{MAX}. (Larger values of L_{MAX} increase memory and performance requirements of the device, and in some cases may increase L.) To estimate the amount of useful information an attacker could collect by monitoring a device's power consumption, for example, a designer might consider the amount of noise in the device's power usage, the power line capacitance, the useful time resolution for power consumption measurements, as well as the strength of the signals being monitored. Similarly, the designer knows that timing measurements can rarely yield more than a few bits of information per operation, since timing information is normally quantized to an integral number of clock cycles. In choosing L_{MAX}, the designer should assume that attackers will be able to combine information gleaned from multiple types of attacks. If the leakage rate is too large (as in the extreme case where L equals the key size because the entire key can be extracted during a single transaction), additional design features should be added to reduce L and reduce the value needed for L_{MAX}. Such additional measures can include known methods, such as filtering the device's power inputs, adding shielding, introducing noise into the timing or power consumption, implementing constant-time and constant execution path algorithms, and changing the device layout. Again, note that the designer of a leak-resistant system does not actually need to know what information is being revealed or how it is leaked; all he or she need do is choose an upper bound for the rate at which attackers might learn information about the keys. In contrast, the designer of a traditional system faces the much harder task of ensuring that no information about the secrets will leak out.

There are many ways information about secrets can leak from cryptosystems. For example, an attacker can use a high-speed analog-to-digital converter to record a smartcard's power consumption during a cryptographic operation. The amount of useful information that can be gained from such a measurement varies, but it would be fairly typical to gain enough information to guess each of 128 key bits correctly with a probability of 0.7. This information can reduce the amount of effort required for a brute force attack. For example, a brute force attack with one message against a key containing k bits where each bit's value is known with probability p can be completed in

$$E(k,p)=\sum_{i=0}^{k}\left[\binom{k}{i}(1-p)^{i}p^{k-i}\left[\left(\sum_{j=0}^{i}\binom{k}{j}\right)-\frac{1}{2}\binom{k}{i}\right]+\frac{1}{2}\right]$$

operations. The reduction in the effort for a brute force attack is equivalent to shortening the key by $L = \log_2(E(k,½) / E(k,p)) = \log_2(k - E(k,p) - 1)$ bits. (For example, in the case of k = 128 and p = 0.7, L is estimated to be about 11 bits for the first measurement. With a multiple message attack, the attacker's effort can fall to as low as $E(k,p)=\frac{1}{p^k}$.) Attackers can gain additional information about the keys by measuring additional operations; unless leak-resistance is used, finding the key becomes easy after just a few dozen operations.

When choosing L_{MAX}, a system designer should consider the signal-to-noise ratio of an attacker's measurements. For example, if the signal and noise are of roughly equivalent magnitude, the designer knows that an attacker's measurements should be incorrect about 25 percent of the time (e.g., p = 0.75 if only one observation per key bit is possible). Many measurement techniques, such as those involving timing, may have signal-to-noise ratios of 1:100 or worse. With such systems, L is generally quite small, but attackers who can make a large number of measurements can use averaging or other statistical techniques to recover the entire key. In extreme cases, attackers may be able to obtain all key bits with virtually perfect accuracy from a single transaction (i.e., $L = k$), necessitating the addition of shielding, noise in the power consumption (or elsewhere), and other measures to reduce p and L. Of course, L_{MAX} should be chosen conservatively; in the example above where less than 4 useful bits are obtained per operation for the given attack, the designer might select $L_{MAX} = 64$ for a leak-proof design.

Leak-proof (and, more generally, leak-resistant) cryptosystems provide system designers with important advantages. When designing a traditional (i.e., non-leak-resistant and non-leak-proof) cryptosystem, a careful cryptosystem designer should study all possible information available to attackers if he or she is to ensure that no analytical techniques could be used to compromise the keys. In practice, many insecure systems are developed and deployed because such analysis is incomplete, too difficult even to attempt, or because the cryptographers working on the system do not understand or cannot completely control the physical characteristics of the device they are designing. Unexpected manufacturing defects or process changes, alterations made to the product by attackers, or modifications made to the product in the field can also introduce problems. Even a system designed and analyzed with great care can be broken if new or improved data collection and analysis techniques are found later. In contrast, with leak-proof cryptography, the system designer only needs to define an upper bound on the maximum rate at which attackers can extract information about the keys. A detailed understanding of the information available to attackers is not required, since leak-proof (and leak-resistant) cryptosystem designs allow for secret information in the device to leak out in (virtually) any way, yet remain secure despite this because leaked information is only of momentary value.

In a typical leak-proof design, with each new cryptographic operation i, the attacker is assumed to be able to choose any function F_i and determine the L_{MAX}-bit result of computing F_i on the device's secrets, inputs, intermediates, and outputs over the course of the operation. The attacker is even allowed to choose a new function F_i with each new operation. The system may be considered leak-proof with a security factor n and leak rate L_{MAX} if, after observing a large number of operations, an attacker cannot forge signatures, decrypt data, or perform other sensitive operations without performing an exhaustive search to find an n-bit key or performing a comparable $O(2^n)$ operation. In addition to choosing L_{MAX}, designers also choose n, and should select a value large enough to make exhaustive search infeasible. In the sections that follow, various embodiments of the invention, as applied to improve the security of common cryptographic operations and protocols, will be described in more detail.

II. Symmetric Cryptographic Protocols

A. Symmetric Authentication

An exemplary cryptographic protocol that can be secured using the techniques of the present invention is symmetric authentication.

1. Conventional Symmetric Authentication

Assume a user wishes to authenticate herself to a server using an n-bit secret key, K, known to both the server and the user's cryptographic token, but not known to attackers. The cryptographic token should be able to resist tampering to prevent, for example, attackers from being able to extract secrets from a stolen token. If the user's token has perfect tamper resistance (i.e., L=0), authentication protocols of the background art can be used. Typically the server sends a unique, unpredictable challenge value R to the user's token, which computes the value $A = H(R \| K)$, where "||" denotes concatenation and H is a one-way cryptographic hash function such as SHA. The user sends A to the server, which independently computes A (using its copy of K) and compares its result with the received value. The user authentication succeeds only if the comparison operation indicates a match.

If the function H is secure and if K is sufficiently large to prevent brute force attacks, attackers should not be able to obtain any useful information from the (R, A) values of old authentication sessions. To ensure that attackers cannot impersonate users by replaying old values of A, the server generates values of R that are effectively (with sufficiently high probability) unique. In most cases, the server should also make R unpredictable to ensure that an attacker with temporary possession of a token cannot compute future values of A. For example, R might be a 128-bit number produced using a secure random number generator (or pseudorandom number generator) in the server. The properties of cryptographic hash functions such as H have been the subject of considerable discussion in the literature, and need not be described in detail here. Hash functions typically provide functionality modeled after a random oracle, deterministically producing a particular output from any input. Ideally, such functions should be collision-resistant, non-invertable, should not leak partial information about the input from the output, and should not leak information about the output unless the entire input is known. Hash functions can have any output size. For example, MD5 produces 128-bit outputs and SHA produces 160-bit outputs. Hash functions may be constructed from other cryptographic primitives or other hash functions.

While the cryptographic security of the protocol using technology of the background art may be good, it is not leak-proof; even a one-bit leak function (with L=1) can reveal the key. For example, if the leak function F equals bit (R mod n) of K, an attacker can break the system quickly since a new key bit is revealed with every transaction where (R mod n) has a new value. Therefore, there is a need for a leak-proof/leak-resistant symmetric authentication protocol.

2. Leak-Resistant Symmetric Authentication

The following is one embodiment of a leak-resistant (and, in fact, also leak-proof) symmetric authentication protocol, described in the context of a maximum leakage rate of L_{MAX} bits per transaction from the token and a security factor n, meaning that attacks of complexity $O(2^n)$, such as brute-force attacks against an n-bit key, are acceptable, but there should not be significantly easier attacks. The user's token maintains a counter t, which is initialized to zero, and an $(n+2L_{MAX})$-bit shared secret K_t, which is initialized with a secret K_0. Note that against adversaries performing precomputation attacks based on Hellman's time/memory trade-off, larger values of n may be in order. Note also that some useful protocol security features, such as user and/or server identifiers in the hash operation inputs, have been omitted for simplicity in the protocol description. It is also assumed that no leaking will occur from the server. For simplicity in the protocol description, some possible security features (such as user and/or server identifiers in the hash operation inputs) have been omitted, and it is assumed that the server is in a physically secure environment. However, those skilled in the art will appreciate that the invention is not limited to such assumptions, which have been made as a matter of convenience rather than necessity.

As in the traditional protocol, the server begins the authentication process by generating a unique and unpredictable value R at step 105. For example, R might be a 128-bit output from a secure random number generator. At step 110, the server sends R to the user's token. At step 112, the token receives R. At step 115, the token increments its counter t by computing $t \leftarrow t + 1$. At step 120, the token updates K_t by computing $K_t \leftarrow H_K(t \parallel K_t)$, where H_K is a cryptographic hash function that produces an $(n+2L_{MAX})$ bit output from the old value of K_t and the (newly incremented) value of t. Note that in the replacement operations (denoted "$\leftarrow$"), the token deletes the old values of t and K_t, replacing them with the new values. By deleting the old K_t, the token ensures that future leak functions cannot reveal information about the old (deleted) value. At step 122, the token uses the new values of t and K_t to compute an authenticator $A = H_A(K_t \| t \| R)$. At step 125, the token sends both t and the authenticator A to the server, which receives them at step 130. At step 135, the server verifies that t is acceptable (e.g., not too large but larger than the value received in the last successful authentication). If t is invalid, the server proceeds to step 175. Otherwise, at step 140, the server initializes its loop counter i to zero and its key register K_t' to K_0. At step 145, the server compares i with the received value of t, proceeding to step 160 if they are equal. Otherwise, at step 150, the server increments i by computing $i \leftarrow i + 1$. At step 155, the server computes $K_t' \leftarrow H_K(i \| K_t')$, then proceeds back to step 145. At step 160, the server computes $A' = H_A(K_t' \| t \| R)$. Finally, at step 165, the server compares A and A', where the authentication succeeds at step 170 if they match, or fails at 175 if they do not match.

This design assumes that at the beginning of any transaction the attacker may have L_{MAX} bits of useful information about the state of the token (e.g., K_t) that were obtained using the leak function F in a previous operation. During the transaction, the attacker can gain an additional L_{MAX} bits of useful information from the token. If, at any time, any $2L_{MAX}$ (or fewer) bits of useful information about the secret are known to the attacker, there are still $(n+2L_{MAX})-2L_{MAX} = n$ or more unknown bits. These n bits of unknown information ensure that attacks will require $O(2^n)$ effort, corresponding to the desired security factor. However, the attacker should have no more than L_{MAX} bits of useful information about K_t at the end of the transaction. The property that attackers lose useful information during normal operation of the system is a characteristic of the leak-proof or leak-resistant cryptosystem. In general, this information loss is achieved when the cryptosystem performs operations that convert attackers' useful partial information about the secret into useless information. (Information is considered useless if it gives an attacker nothing better than the ability to test candidate values in an $O(2^n)$ exhaustive search or other "hard" operation. For example, if exhaustive search of X is hard and H is a good hash function, $H(X)$ is useless information to an attacker trying to find X.)

Thus, the attacker is assumed to begin with L_{MAX} bits of useful information about K_t before the token's $K_t \leftarrow H_K(t \| K_t)$ computation. (Initial information about anything other than K_t is of no value to an attacker because K_t is the only secret value in the token. The function H_K and the value of t are not assumed to be secret.) The attacker's information can be any function of K_t produced from the previous operation's leaks.

3. Security Characteristics of Leak-Proof Systems

The following section provides a technical discussion of the security characteristics of the exemplary leak-proof system described above. The following analysis is provided as an example of how the design can be analyzed, and how a system may be designed using general assumptions about attackers' capabilities. The discussion and assumptions do not necessarily apply to other embodiments of the invention and should not be construed as limiting the scope or applicability of the invention in any way.

During the course of a transaction, the leak function F might reveal up to L_{MAX} information about the system and its secrets. The design assumes that any information contained in the system may be leaked by F, provided that F does not reveal useful new information about values of K_t that were deleted before the operation started, and F does not reveal useful information about values of K_t that will be computed in future operations. These constraints are completely reasonable, since real-world leaks would not reveal information about deleted or not-yet-existent data. (The only way information about future K_t values could be leaked would be the bizarre case where the leak function itself included, or was somehow derived from, the function H_K.) In practice, these constraints on F are academic and of little concern, but they are relevant when constructing proofs to demonstrate the security of a leak-proof system.

If the leak occurs at the beginning of the H_K computation, it could give the attacker up to $2L_{MAX}$ bits of useful information about the input value of K_t. Because K_t contains $(2L_{MAX}+n)$ bits of secret information and the attacker may have up to $2L_{MAX}$ bits of useful information about the initial value of K_t, there remain at least $(2L_{MAX}+n)-2L_{MAX} = n$ bits of information in K_t that are secret. The hash function H_K effectively mixes up these n bits to produce a secure new K_t during each transaction such that the attacker's information about the old K_t is no longer useful.

If the leak occurs at the end of the H_K computation, it could give an attacker up to L_{MAX} bits of information about the final value of H_K, yielding L_{MAX} bits of information about the input to the subsequent transaction. This is not a problem, since the design assumes that attackers have up to L_{MAX} bits of information about K_t at the beginning of each transaction.

A third possibility is that the attacker's L_{MAX} bits of information might describe intermediates computed during the operation H_K. However, even if the attacker could obtain L_{MAX} new bits of information about the input to H_K and also L_{MAX} bits of information about the output from H_K, the system would be secure, since the attacker would never have more than $2L_{MAX}$ bits of information about the input K_t or more than L_{MAX} bits of information about the output K_t. Provided that L_{MAX} bits of information from within H_K cannot reveal more than L_{MAX} bits of information about the input, or more than L_{MAX} bits of information about the output, the system will be secure. This will be true unless H_K somehow compresses the input to form a short intermediate which is expanded to form the output. While hash functions whose internal states are smaller than their outputs should not be used, most cryptographic hash functions are fine.

A fourth possibility is that part or all of the leak could occur during the $A = H_A(K_t \| t \| R)$ calculation. The attacker's total "budget" for observations is L_{MAX} bits. If L_1 bits of leak occur during the H_K computation, an additional L_2 bits of information can leak during the $A = H_A(K_t \| t \| R)$ operation, where $L_2 \leq L_{MAX} - L_1$. If the second leak provides information about K_t, this is no different from leaking information about the result of the H_K computation; the attacker will still conclude the transaction with no more than L_{MAX} bits of information about K_t because $L_1 + L_2 \leq L_{MAX}$. However, the second leak could reveal information about A. To keep A secure against leaks (to prevent, for example, an attacker from using a leak to capture A and using A before the legitimate user can), the size of A should include an extra L_{MAX} bits (to provide security even if $L_2 = L_{MAX}$). Like H_K, H_A should not leak information about deleted or future values of K_t that are not used in or produced by the given operation. As with the similar assumptions on leaks from H_K, this limitation is primarily academic and of little practical concern, since real-world leak functions do not reveal information about deleted or not-yet-computed data. However, designers might be cautious when using unusual designs for H_A that are based on or derived from H_K, particularly if the operation $H_A(K_t \| t \| R)$ could reveal useful information about the result of computing $H_K(t \| K_t)$.

B. Other Leak-Resistant Symmetric Schemes

The same basic technique of updating a key (K) with each transaction, such that leakage about a key during one transaction does not reveal useful information about a key in a subsequent (or past) transaction, can be easily extended to other applications besides authentication.

1. Symmetric Data Verification

For example and without limitation, leak-resistant symmetric data verification is often useful where a device needs to support symmetrically-signed code, data, content, or parameter updates (all of which will, as a matter of convenience, be denoted as "data" herein). In existing systems, a hash or MAC of the data is typically computed using a secret key and the data is rejected if computed hash or MAC does not match a value received with the data. For example, a MAC may be computed as HMAC(K, data), where HMAC is defined in "RFC 2104, HMAC: Keyed-Hashing for Message Authentication" by H. Krawczyk, M. Bellare, and R. Canetti, 1997. Traditional (non-leak-resistant) designs are often vulnerable to attacks including power consumption analysis of MAC functions and timing analysis of comparison operations.

In an exemplary leak-resistant verification protocol, a verifying device (the "verifier") maintains a counter t and a key K_t, which are initialized (for example at the factory) with $t \leftarrow 0$ and $K_t \leftarrow K_0$. Before the transaction, the verifier provides t to the device providing the signed data (the "signer"), which also knows K_0. The signer uses t to compute K_{t+1}' (the prime indicating a quantity derived by the signer, rather than at the verifier) from K_0 (or K_t' or any other available value of K_i'). using the relation $K_i' = H_K(i \parallel K_{i-1}')$, computes signature $S' = \mathrm{HMAC}(K_{t+1}'$, data), and sends S' plus any other needed information (such as data or t) to the verifier. The verifier confirms that the received value of t (if any) matches its value of t, and rejects the signature if it does not. If t matches, the verifier increments t and updates K_t in its nonvolatile memory by computing $t \leftarrow t + 1$ and $K_t \leftarrow H_K(t \parallel K_t)$. In an alternative embodiment, if the received value of t is larger than the internal value but the difference is not unreasonably large, it may be more appropriate to accept the signature and perform multiple updates to K_t (to catch up with the signer) instead of rejecting the signature outright. Finally, the verifier computes $S = \mathrm{HMAC}(K_t$, data) and verifies that $S = S'$, rejecting the signature if S does not equal the value of S' received with the data.

2. Symmetric Encryption

Besides authentication and verification, leak-resistant symmetric cryptography can also be tailored to a wide variety of applications and environments. For example, if data encryption is desired instead of authentication, the same techniques as were disclosed above may be used to generate a key K_t used for encryption rather than verification.

3. Variations in Computational Implementation

In the foregoing, various applications were disclosed for the basic technique of updating a key K_t in accordance with a counter and deleting old key values to ensure that future leakage cannot reveal information about the now-deleted key. Those skilled in the art will realize, however, that the exemplary techniques described above may be modified in various ways without departing from the spirit and scope of the invention. For example, if communications between the device and the server are unreliable (for example if the server uses voice recognition or manual input to receive t and A), then small errors in the signature may be ignored. (One skilled in the art will appreciate that many functions may be used to determine whether a signature corresponds – sufficiently closely -- to its expected value.) In another variation of the basic technique, the order of operations and of data values may be adjusted, or additional steps and parameters may be added, without significantly changing the invention. In another variation, to save on communication bandwidth or memory, the high order bits or digits of t may not need to be communicated or remembered. In another variation, as a performance optimization, devices need not recompute K_t from K_0 with each new transaction. For example, when a transaction succeeds, the server can discard K_0 and maintain the validated version of K_t. In another variation, if bi-directional authentication is required, the protocol can include a step whereby the server can authenticates itself to the user (or user's token) after the user's authentication is complete. In another variation, if the server needs to be secured against leaks as well (as in the case where the role of "server" is played by an ordinary user), it can maintain its own counter t. In each transaction, the parties agree to use the larger of their two t values, where the device with the smaller t value performs extra updates to K_t to synchronize t. In an alternate embodiment for devices that contain a clock and a reliable power source (e.g., battery), the update operation may be performed periodically, for example by computing $K_t \leftarrow H_K(t \| K_t)$ once per second. The token uses the current K_t to compute $A = H_A(K_t \| t \| R)$ or, if the token does not have any means for receiving R, it can output $A = H_A(K_t)$. The server can use its clock and local copy of the secret to maintain its own version of K_t, which it can use to determine whether received values of A are recent and correct. All of the foregoing show that the method and apparatus of the present invention can be implemented using numerous variations and modifications to the exemplary embodiments described herein, as would be understood by one skilled in the art.

III. Asymmetric Cryptographic Protocols

The foregoing illustrates various embodiments of the invention that may be used with symmetric cryptographic protocols. As will be seen below, still other techniques of the present invention may be used in connection with asymmetric cryptographic operations and protocols. While symmetric cryptosystems are sufficient for some applications, asymmetric cryptography is required for many applications. There are several ways leak resistance can be incorporated into public key cryptosystems, but it is often preferable to have as little impact as possible on the overall system architecture. Most of the exemplary designs have thus been chosen to incorporate leak resistance into widely used cryptosystems in a way that only alters the key management device, and does not affect the certification process, certificate format, public key format, or processes for using the public key.

A. Certified Diffie-Hellman

Diffie-Hellman exponential key exchange is a widely used asymmetric protocol whereby two parties who do not share a secret key can negotiate a shared secret key. Implementations of Diffie-Hellman can leak information about the secret exponents, enabling attackers to determine the secret keys produced by those implementations. Consequently, a leak-resistant implementation of Diffie-Hellman would be useful. To understand such a leak-resistant implementation, it will be useful to first review a conventional Diffie-Hellman implementation.

1. Conventional Certified Diffie-Hellman

Typical protocols in the background art for performing certified Diffie-Hellman exponential key agreement involve two communicating users (or devices) and a certifying authority (CA). The CA uses an asymmetric signature algorithm (such as DSA) to sign certificates that specify a user's public Diffie-Hellman parameters (the prime p and generator g), public key (p^x mod g, where x is the user's secret exponent), and auxiliary information (such as the user's identity, a description of privileges granted to the certificate holder, a serial number, expiration date, etc.). Certificates may be verified by anyone with the CA's public signature verification key. To obtain a certificate, user U typically generates a secret exponent (x_u), computes his or her own public key $y_u = g^{x_u} \bmod p$, presents y_u along with any required auxiliary identifying or authenticating information (e.g., a passport) to the CA, who issues the user a certificate C_U. Depending on the system, p and g may be unique for each user, or they may be system-wide constants (as will be assumed in the following description of Diffie-Hellman using the background art).

Using techniques of the background art, Alice and Bob can use their certificates to establish a secure communication channel. They first exchange certificates (C_{Alice} and C_{Bob}). Each verifies that the other's certificate is acceptable (e.g., properly formatted, properly signed by a trusted CA, not expired, not revoked, etc.). Because this protocol will assume that p and g are constants, they also check that the certificate's p and g match the expected values. Alice extracts Bob's public key (y_{Bob}) from C_{Bob} and uses her secret exponent (x_{Alice}) to compute $z_{\text{Alice}} = (y_{\text{Bob}})^{x_{\text{Alice}}} \bmod p$. Bob uses his secret exponent and Alice's public key to compute $z_{\text{Bob}} = (y_{\text{Alice}})^{x_{\text{Bob}}} \bmod p$. If everything works correctly, $z_{\text{Alice}} = z_{\text{Bob}}$, since:

$$\begin{aligned} z_{\text{Alice}} &= (y_{\text{Bob}})^{x_{\text{Alice}}} \bmod p \\ &= (g^{x_{\text{Bob}}})^{x_{\text{Alice}}} \bmod p \\ &= (g^{x_{\text{Alice}}})^{x_{\text{Bob}}} \bmod p \\ &= (y_{\text{Alice}})^{x_{\text{Bob}}} \bmod p \\ &= z_{\text{Bob}}. \end{aligned}$$

Thus, Alice and Bob have a shared key $z = z_{\text{Alice}} = z_{\text{Bob}}$. An attacker who pretends to be Alice but does not know her secret exponent (x_{Alice}) will not be able to compute $z_{\text{Alice}} = (y_{\text{Bob}})^{x_{\text{Alice}}} \bmod p$ correctly. Alice and Bob can positively identify themselves by showing that they correctly found z. For example, each can compute and send the other the hash of z concatenated with their own certificate. Once Alice and Bob have verified each other, they can use a symmetric key derived from z to secure their communications. (For an example of a protocol in the background art that uses authenticated Diffie-Hellman, see "The SSL Protocol Version 3.0" by A. Freier, P. Karlton, and P. Kocher, March 1996.)

2. Leak-Resistant Certified Diffie-Hellman

A satisfactory leak-resistant public key cryptographic scheme should overcome the problem that, while certification requires the public key be constant, information about the corresponding private key should not leak out of the token that contains it. In the symmetric protocol described above, the design assumes that the leak function reveals no useful information about old deleted values of K_t or about future values of K_t that have not yet been computed. Existing public key schemes, however, require that implementations repeatedly perform a consistent, usually deterministic, operation using the private key. For example, in the case of Diffie-Hellman, a leak-resistant token that is compatible with existing protocols and implementations should be able to perform the secret key operation $y^x \bmod p$, while ensuring that the exponent x remains secret. The radical reshuffling of the secret provided by the hash function H_K in the symmetric approach cannot be used because the device should be able to perform the same operation consistently.

The operations used by the token to perform the private key operation are modified to add leak resistance using the following variables:

Register	Comment
x_1	First part of the secret key (in nonvolatile updateable memory)
x_2	Second part of the secret key (in nonvolatile updateable memory)
g	The generator (not secret).
p	The public prime, preferably a strong prime (not secret).

The prime p and generator g may be global parameters, or may be specific to individual users or groups of users (or tokens). In either case, the certificate recipient should be able to obtain p and g securely, usually as built-in constants or by extracting them from the certificate.

To generate a new secret key, the key generation device (often but not always the cryptographic token that will contain the key) first obtains or generates p and g, where p is the prime and g is a generator mod p. If p and g are not system-wide parameters, algorithms known in the background art for selecting large prime numbers and generators may be used. It is recommended that p be chosen with $\frac{p-1}{2}$ also prime, or at least that $\phi(p)$ not be smooth. (When $\frac{p-1}{2}$ is not prime, information about x_1 and x_2 modulo small factors of $\phi(p)$ may be leaked, which is why it is preferable that $\phi(p)$ not be smooth. Note that ϕ denotes Euler's totient function.) Once p and g have been chosen, the device generates two random exponents x_1 and x_2. The lowest-order bit of x_1 and of x_2 is not considered secret, and may be set to 1. Using p, g, x_1, and x_2, the device can then compute its public key as $g^{x_1 x_2} \bmod p$ and submit it, along with any required identifying information or parameters needed (e.g., p and g), to the CA for certification.

Figure 2 illustrates the process followed by the token to perform private key operations. At step 205, the token obtains the input message y, its own (non-secret) prime p, and its own secret key halves (x_1 and x_2). If x_1, x_2, and p are stored in encrypted and/or authenticated form, they would be decrypted or verified at this point. At this step, the token should verify that $1 < y < p-1$. At step 210, the token uses a random number generator (or pseudorandom number generator) to select a random integer b_0, where $0 < b_0 < p$. At step 215, the token computes $b_1 = b_0^{-1} \bmod p$. The inverse computation mod p may be performed using the extended Euclidean algorithm or the formula $b_1 = b_0^{\phi(p)-1} \bmod p$. At step 220, the token computes $b_2 = b_1^{x_1} \bmod p$. At this point, b_1 is no longer needed; its storage space may be used to store b_2. Efficient algorithms for computing modular exponentiation, widely known in the art, may be used to complete step 220. Alternatively, when a fast modular exponentiator is available, the computation b_2 may be performed using the relationship $b_2 = b_0^{\phi(p)-x_1} \bmod p$. At step 225, the token computes $b_3 = b_2^{x_2} \bmod p$. At this point, b_2 is no longer needed; its storage space may be used to store b_3. At step 230, the token computes $z_0 = b_0 y \bmod p$. At this point, y and b_0 are no longer needed; their space may be used to store r_1 (computed at step 235) and z_0. At step 235, the token uses a random number generator to select a random integer r_1, where $0 < r_1 < \phi(p)$ and $\gcd(r_1, \phi(p)) = 1$. (If $\frac{p-1}{2}$ is known to be prime, it is sufficient to verify that r_1 is odd.) At step 240, the token updates x_1 by computing $x_1 \leftarrow x_1 r_1 \bmod \phi(p)$. The old value of x_1 is deleted and replaced with the updated value. At step 245, the token computes $r_2 = (r_1^{-1}) \bmod \phi(p)$. If $\frac{p-1}{2}$ is prime, then r_2 can be found using a modular exponentiator and the Chinese Remainder Theorem. Note that r_1 is not needed after this step, so its space may be used to store r_2. At step 250, the token updates x_2 by computing $x_2 \leftarrow x_2 r_2 \bmod \phi(p)$. The old value of x_2 should be deleted and replaced with the updated value. At step 255, the token computes $z_1 = (z_0)^{x_1} \bmod p$. Note that z_0 is not needed after this step, so its space may be used to store z_1. At step 260, the token computes $z_2 = (z_1)^{x_2} \bmod p$. Note that z_1 is not needed after this step, so its space may be used to store z_2. At step 265, the token finds the exponential key exchange result by computing $z = z_2 b_3 \bmod p$. Finally, at step 270, the token erases and frees any remaining temporary variables.

The process shown in Figure 2 correctly computes $z = y^x \bmod p$, where $x = x_1 x_2 \bmod \phi(p)$, since:

$$\begin{aligned}
z &= z_2 b_3 \bmod p \\
&= \left(z_1^{x_2} \bmod p\right)\left(b_2^{x_2} \bmod p\right) \bmod p \\
&= \left(\left(z_0^{x_1} \bmod p\right)^{x_2}\right)\left(\left(b_1^{x_1} \bmod p\right)^{x_2}\right) \bmod p \\
&= \left(b_0 y \bmod p\right)^{x_1 x_2}\left(b_0^{-1} \bmod p\right)^{x_1 x_2} \bmod p \\
&= y^{x_1 x_2} \bmod p \\
&= y^{x} \bmod p.
\end{aligned}$$

The invention is useful for private key owners communicating with other users (or devices) who have certificates, and also when communicating with users who do not.

If Alice has a certificate and wishes to communicate with Bob who does not have a certificate, the protocol proceeds as follows. Alice sends her certificate (C_{Alice}) to Bob, who receives it and verifies that it is acceptable. Bob extracts y_{Alice} (along with p_{Alice} and g_{Alice}, unless they are system-wide parameters) from C_{Alice}. Next, Bob generates a random exponent x_{BA}, where $0 < x_{\text{AB}} < \phi(p_{\text{Alice}})$. Bob then uses his exponent x_{AB} and Alice's parameters to calculate $y_{\text{BA}} = \left(g_{\text{Alice}}^{x_{\text{BA}}}\right) \bmod p_{\text{Alice}}$ and the session key $z = \left(y_{\text{Alice}}^{x_{\text{BA}}}\right) \bmod p_{\text{Alice}}$. Bob sends y_{BA} to Alice, who performs the operation illustrated in Figure 2 to update her internal parameters and derive z from y_{BA}. Alice then proves that she computed z correctly, for example by sending Bob $H(z \parallel C_{\text{Alice}})$. (Alice cannot authenticate Bob because he does not have a certificate. Consequently, she does not necessarily need to verify that he computed z successfully.) Finally, Alice and Bob can use z (or, more commonly, a key derived from z) to secure their communications.

If both Alice and Bob have certificates, the protocol works as follows. First, Alice and Bob exchange certificates (C_{Alice} and C_{Bob}), and each verifies that other's certificate is valid. Alice then extracts the parameters p_{Bob}, g_{Bob}, and y_{Bob} from C_{Bob}, and Bob extracts p_{Alice}, g_{Alice}, and y_{Alice} from C_{Alice}. Alice then generates a random exponent x_{AB} where $0 < x_{\text{AB}} < \phi(p_{\text{Bob}})$, computes $y_{\text{AB}} = \left(g_{\text{Bob}}\right)^{x_{\text{AB}}} \bmod p_{\text{Bob}}$, and computes $z_{\text{AB}} - \left(y_{\text{Bob}}\right)^{x_{\text{AB}}} \bmod p_{\text{Bob}}$. Bob generates a random x_{BA} where $0 < x_{\text{BA}} < \phi(p_{\text{Alice}})$, computes $y_{\text{BA}} = \left(g_{\text{Alice}}\right)^{x_{\text{BA}}} \bmod p_{\text{Alice}}$, and computes $z_{\text{BA}} - \left(y_{\text{Alice}}\right)^{x_{\text{BA}}} \bmod p_{\text{Alice}}$. Bob sends y_{BA} to Alice, and Alice sends y_{AB} to Bob. Alice and Bob each perform the operation shown in Figure 2, where each uses the prime p from their own certificate and their own secret exponent halves (x_1 and x_2). For the message y in Figure 2, Alice uses y_{BA} (received from Bob), and Bob uses y_{AB} (received from Alice). Using the process shown in Figure 2, Alice computes z. Using z and z_{AB} (computed previously), she can find a session key K. This may be done, for example, by using a hash function H to compute $K = \mathrm{H}(z \| z_{AB})$. The value of z Bob obtains using the process shown in Figure 2 should equal Alice's z_{AB}, and Bob's z_{BA} (computed previously) should equal Alice's z. If there were no errors or attacks, Bob should thus be able to find K, e.g., by computing $K = \mathrm{H}(z_{BA} \| z)$. Alice and Bob now share K. Alice can prove her identity by showing that she computed K correctly, for example by sending Bob $\mathrm{H}(K \| C_{Alice})$. Bob can prove his identity by sending Alice $\mathrm{H}(K \| C_{Bob})$. Alice and Bob can then secure their communications by encrypting and authenticating using K or a key derived from K.

Note that this protocol, like the others, is provided as an example only; many variations and enhancements of the present invention are possible and will be evident to one skilled in the art. For example, certificates may come from a directory, more than two parties can participate in the key agreement, key escrow functionality may be added, the prime modulus p may be replaced with a composite number, etc. Note also that Alice and Bob as they are called in the protocol are not necessarily people; they would normally be computers, cryptographic devices, etc.

For leak resistance to be effective, attackers should not be able to gain new useful information about the secret variables with each additional operation unless a comparable amount of old useful information is made useless. While the symmetric design is based on the assumption that leaked information will not survive the hash operation H_K, this design uses multiplication operations mod $\phi(p)$ to update x_1 and x_2. The most common variety of leaked information, statistical information about exponent bits, is not of use to attackers in this design, as the exponent update process ($x_1 \leftarrow x_1 r_1 \bmod \phi(p)$ and $x_2 \leftarrow x_2 r_2 \bmod \phi(p)$) destroys the utility of this information. The only relevant characteristic that survives the update process is that $x_1 x_2 \bmod \phi(p)$ remains constant, so the system designer should be careful to ensure that the leak function does not reveal information allowing the attacker to find new useful information about $x_1 x_2 \bmod \phi(p)$.

There is a modest performance penalty, approximately a factor of four, for the leak-resistant design as described. One way to improve performance is to remove the blinding and unblinding operations, which are often unnecessary. (The blinding operations prevent attackers from correlating input values of y with the numbers processed by the modular exponentiation operation.) Alternatively or additionally, it is possible to update and reuse values of b_0, b_3, r_1, and r_2 by computing $b_0 \leftarrow (b_0)^v \bmod p$, $b_3 \leftarrow (b_3)^v \bmod p$, $r_1 \leftarrow (r_1)^w \bmod \phi(p)$, and $r_2 \leftarrow (r_2)^w \bmod \phi(p)$, where v and w are fairly short random exponents. Note that the relationship $b_3 \leftarrow b_0^{-x_1 x_2} \bmod p$ remains true when b_0 and b_3 are both raised to the power v (mod p). The relationship $r_2 = (r_1^{-1}) \bmod \phi(p)$ also remains true when r_1 and r_2 are exponentiated (mod $\phi(p)$). Other parameter update operations may also be used, such as exponentiation with fixed exponents (e.g., $v = w = 3$), or multiplication with random values and their inverses, mod p and $\phi(p)$. The time per transaction with this update process is about half that of the unoptimized leak-resistant implementation, but additional storage is required and care should be taken to ensure that b_0, b_3, r_1, and r_2 will not be leaked or otherwise compromised.

It should also be noted that with this particular type of certified Diffie-Hellman, the negotiated key is the same every time any given pair of users communicate. Consequently, though the blinding operation performed using b_0 and b_3 does serve to protect the exponents, the result K can be leaked in the final step or by the system after the process is complete. If storage is available, parties could keep track of the values of y they have received (or their hashes) and reject duplicates. Alternatively, to ensure that a different result is obtained from each negotiation, Alice and Bob can generate and exchange additional exponents, w_{Alice} and w_{Bob}, for example with $0 < w < 2^{128}$ (where $2^{128} << p$). Alice sets $y = (y_{\text{BA}})^{w_{\text{Alice}} w_{\text{Bob}}} \bmod p$ instead of just $y = y_{\text{BA}}$, and Bob sets $y = (y_{\text{AB}})^{w_{\text{Bob}} w_{\text{Alice}}} \bmod p$ instead of $y = y_{\text{AB}}$ before performing the operation shown in Figure 2.

B. Leak-Resistant RSA

Another asymmetric cryptographic protocol is RSA, which is widely used for digital signatures and public key encryption. RSA private key operations rely on secret exponents. If information about these secret exponents leaks from an implementation, its security can be compromised. Consequently, a leak-resistant implementation of RSA would be useful.

To give RSA private key operations resistance to leaks, it is possible to divide the secret exponent into two halves such that information about either half is destroyed with each operation. These are two kinds of RSA private key operations. The first, private key signing, involves signing a message with one's own private key to produce a digital signature verifiable by anyone with one's corresponding public key. RSA signing operations involve computing $S = M^d \bmod n$, where M is the message, S is the signature (verifiable using $M = S^e$

mod n), d is the secret exponent and equals e^{-1} mod $\phi(n)$, and n is the modulus and equals pq, where n and e are public and p and q are secret primes, and ϕ is Euler's phi function. An RSA public key consists of e and n, while an RSA private key consists of d and n (or other representations of them). For RSA to be secure, d, $\phi(n)$, p, and q should all be secret.

The other RSA operation is decryption, which is used to recover messages encrypted using one's public key. RSA decryption is virtually identical to signing, since the decrypted message M is recovered from the ciphertext C by computing $M = C^d \bmod n$, where the ciphertext C was produced by computing $C=M^e \bmod n$. Although the following discussion uses variable names from the RSA signing operation, the same techniques may be applied similarly to decryption.

An exemplary leak-resistant scheme for RSA implementations may be constructed as illustrated in Figure 3. At step 300, prior to the commencement of any signing or decryption operations, the device is initialized with (or creates) the public and private keys. The device contains the public modulus n and the secret key components d_1, d_2, and z, and k, where k is a prime number of medium-size (e.g., $0 < k < 2^{128}$) chosen at random, $z = k\phi(n)$, d_1 is a random number such that $0 < d_1 < z$ and $\gcd(d_1, z) = 1$, and $d_2 = (e^{-1} \bmod \phi(n))(d_1^{-1} \bmod z) \bmod z$. In this invention, d_1 and d_2 replace the usual RSA secret exponent d. Techniques for generating the initial RSA primes (e.g., p and q) and modulus (n) are well known in the background art. At step 305, the device computes a random prime k' of medium size (e.g., $0 < k' < 2^{128}$). (Algorithms for efficiently generating prime numbers are known in the art.)

At step 303, the device (token) receives a message M to sign (or to decrypt). At step 310, the device updates z by computing $z \leftarrow k'z$. At step 315, the device updates z again by computing $z \leftarrow z / k$. (There should be no remainder from this operation, since k divides z.) At step 320, k is replaced with k' by performing $k \leftarrow k'$. Because k' will not be used in subsequent operations, its storage space may be used to hold R (produced at step 325). At step 325, the device selects a random R where $0 < R < z$ and $\gcd(R, z) = 1$. At step 330, the device updates d_1 by computing $d_1 \leftarrow d_1 R \bmod z$. At step 335, the device finds the inverse of R by computing $R' \leftarrow R^{-1} \bmod z$ using, for example, the extended Euclidean algorithm. Note that R is no longer needed after this step, so its storage space may be erased and used to hold R'. At step 340, the device updates d_2 by computing $d_2 \leftarrow d_2 R' \bmod z$. At step 345, the device computes $S_0 = M^{d_1} \bmod n$, where M is the input message to be signed (or the message to be decrypted). Note that M is no longer needed after this step, so its storage space may be used for S_0. At step 350, the device computes $S = S_0^{d_2} \bmod n$, yielding the final signature (or plaintext if decrypting a message). Leak-resistant RSA has similar security characteristics as normal RSA; standard message padding, post-processing, and key sizes may be used. Public key operations are also performed normally (e.g., $M = S^e \bmod n$).

A simpler RSA leak resistance scheme may be implemented by splitting the exponent d into two halves d_1 and d_2 such that $d_1 + d_2 = d$. This can be achieved during key generation by choosing d_1 to be a random integer where $0 \leq d_1 \leq d$, and choosing $d_2 \leftarrow d - d_1$. To perform private key operations, the device needs d_1 and d_2, but it does not need to contain d. Prior to each private key operation, the cryptographic device identifies which of d_1 and d_2 is larger. If $d_1 > d_2$, then the device computes a random integer r where $0 \leq r \leq d_1$, adds r to d_2 (i.e., $d_2 \leftarrow d_2 + r$), and subtracts r from d_1 (i.e., $d_1 \leftarrow d_1 - r$). Otherwise, if $d_1 \leq d_2$, then the device chooses a random integer r where $0 \leq r \leq d_2$, adds r to d_1 (i.e., $d_1 \leftarrow d_1 + r$), and subtracts r from d_2 (i.e., $d_2 \leftarrow d_2 - r$). Then, to perform the private key operation on a message M, the device computes $s_1 = M^{d_1} \bmod n$, $s_2 = M^{d_2} \bmod n$, and computes the signature $S = s_1 s_2 \bmod n$. While this approach of splitting the exponent into two halves whose sum equals the exponent can also be used with Diffie-Hellman and other cryptosystems, dividing the exponent into the product of two numbers mod $\phi(p)$ is usually preferable since the assumption that information about d_1+d_2 will not leak is less conservative than the assumption that information about $x_1x_2 \bmod \phi(p)$ will not leak. In the case of RSA, updates mod $\phi(n)$ cannot be done safely, since $\phi(n)$ must be kept secret.

When the Chinese Remainder Theorem is required for performance, it is possible to use similar techniques to add leak resistance by maintaining multiples of the secret primes (p and q) that are updated every time (e.g., multiplying by the new multiple then dividing by the old multiple). These techniques also protect the exponents (d_p and d_q) as multiples of their normal values. At the end of the operation, the result S is corrected to compensate for the adjustments to d_p, d_q, p, and q.

An exemplary embodiment maintains state information consisting of the values n, B_i, B_f, k, p_k, q_k, d_{pk}, d_{qk}, p_{Inv}, and f. To convert a traditional RSA CRT private key (consisting of p, q, d_p, and d_q with $p < q$) into the new representation, a random value for k is chosen, where $0 < k < 2^{64}$. The value B_i is chosen at random where $0 < B_i < n$, and R_1 and R_2 are chosen at random where $0 < R_1 < 2^{64}$ and $0 < R_2 < 2^{64}$. (Of course, constants such as 2^{64} are chosen as example values. It is possible, but not necessary, to place constraints on random numbers, such as requiring that they be prime.) The leak-resistant private key state is then initialized by setting $n \leftarrow pq$, $B_f \leftarrow B_i^{-d} \bmod n$, $p_k \leftarrow (k)(p)$, $q_k \leftarrow (k)(q)$, $d_{pk} \leftarrow d_p + (R_1)(p) - R_1$, $d_{qk} \leftarrow d_q + (R_2)(q) - R_2$, $p_{Inv} \leftarrow k(p^{-1} \bmod q)$, and $f \leftarrow 0$.

To update the system state, first a random value α may be produced where $0 < \alpha < 2^{64}$. Then compute $p_k \leftarrow ((\alpha)(p_k)) / k$, $q_k \leftarrow ((\alpha)(q_k)) / k$, $p_{Inv} \leftarrow ((\alpha)(p_{Inv})) / k$, $k \leftarrow \alpha$. The exponents d_{pk} and d_{qk} may be updated by computing $d_{pk} \leftarrow d_{pk} \pm (R_3 p_k - R_3 k)$ and $d_{qk} \leftarrow d_{qk} \pm (R_4 q_k - R_4 k)$, where R_3 and R_4 can be random or constant values (even 1). The blinding factors B_i and B_f may be updated by computing $B_i = B_i^2 \bmod n$ and $B_f = B_f^2 \bmod n$, by computing new blinding factors, by exponentiating with a value other than 2, etc. Update processes should be performed as often as practical, for example before or after each modular exponentiation process. Before the update begins, a failure counter f is incremented, and when the update completes f is set to zero. If f ever exceeds a threshold value indicating too many consecutive failures, the device should temporarily or permanently disable itself. Note that if the update process is interrupted, memory values should not be left in intermediate states. This can be done by using complete reliable memory updates. If the total set of variable changes is too large for a single complete update, it is possible to store α first then do each variable update reliably which keeping track of how many have been completed.

To perform a private key operation (such as decryption or signing), the input message C is received by the modular exponentiator. Next, the value is blinded by computing $C' \leftarrow (C)(B_i) \bmod n$. The blinded input message is then used to compute modified CRT intermediates by computing $m_{pk} \leftarrow (C')^{d_{pk}} \bmod p_k$ and $m_{qk} \leftarrow (C')^{d_{qk}} \bmod q_k$. Next in the exemplary embodiment, the CRT intermediates are multiplied by k, e.g. $m_{pk} \leftarrow (k)(m_{pk}) \bmod p_k$ and $m_{qk} \leftarrow (k)(m_{qk}) \bmod q_k$. The CRT difference is then computed as $m_{pqk} = (m_{pk} [+ q_k] - m_{qk}) [\bmod q_k]$, where the addition of q_k and/or reduction mod q_k are optional. (The addition of q_k ensures that the result is non-negative.) The blinded result can be computed as

$$M' = \frac{(m_{pk})k + p_k\left[\left(\frac{(p_{Inv})(m_{pqk})}{k}\right) \bmod q_k\right]}{k^2},$$

then the final result M is computed as $M = (M')B_f \bmod n$.

As one of ordinary skill in the art will appreciate, variant forms of the invention are possible. For example, the computational processes can be re-ordered or modified without significantly changing the invention. Some portions (such as the initial and blinding steps) can be skipped. In another example, it is also possible to use multiple blinding factors (for example, instead of or in addition to the value k).

In some cases, other techniques may also be appropriate. For example, exponent vector codings may be rechosen frequently using, for example, a random number generator. Also, Montgomery arithmetic may be performed mod j where j is a value that is changed with each operation (as opposed to traditional Montgomery implementations where j is constant with $j = 2^k$). The foregoing shows that the method and apparatus of the present invention can be implemented using numerous variations and modifications to the exemplary embodiments described herein, as would be known by one skilled in the art.

C. Leak-Resistant ElGamal Public Key Encryption and Digital Signatures

Still other asymmetric cryptographic protocols that may be improved using the techniques of the invention. For example, ElGamal and related cryptosystems are widely used for digital signatures and public key encryption. If information about the secret exponents and parameters leaks from an ElGamal implementation, security can be compromised. Consequently, leak-resistant implementations of ElGamal would be useful.

The private key in the ElGamal public key encryption scheme is a randomly selected secret a where $1 \leq a \leq p-2$. The non-secret parameters are a prime p, a generator α, and $\alpha^a \bmod p$. To encrypt a message m, one selects a random k (where $1 \leq k \leq p-2$) and computes the ciphertext (γ, δ) where $\gamma = \alpha^k \bmod p$ and $\delta = m(\alpha^a \bmod p)^k \bmod p$. Decryption is performed by computing $m = \delta(\gamma^{p-1-a}) \bmod p$. (See the Handbook of Applied Cryptography by A. Menezes, P. van Oorschot, and S. Vanstone, 1997, pages 294-298, for a description of ElGamal public-key encryption).

To make the ElGamal public-key decryption process leak-resistant, the secret exponent $(p - 1 - a)$ is stored in two halves a_1 and a_2, such that $a_1 a_2 = (\phi(p) - a) \bmod \phi(p)$. When generating ElGamal parameters for this leak-resistant implementation, it is recommended, but not required, that p be chosen with $\frac{p-1}{2}$ prime so that $\phi(p)/2$ is prime. The variables a_1 and a_2 are normally chosen initially as random integers between 0 and $\phi(p)$.

Alternatively, it is possible to generate a first, then choose a_1 and a_2, as by selecting a_1 relatively prime to $\phi(p)$ and computing $a_2 = (a^{-1} \bmod \phi(p))(a_1^{-1} \bmod \phi(p)) \bmod \phi(p)$.

Figure 4 illustrates an exemplary leak-resistant ElGamal decryption process. At step 405, the decryption device receives an encrypted message pair (γ, δ). At step 410, the device selects a random r_1 where $1 \leq r_1 < \phi(p)$ and $\gcd(r_1, \phi(p)) = 1$. At step 415, the device updates a_1 by computing $a_1 \leftarrow a_1 r_1 \bmod \phi(p)$, over-writing the old value of a_1 with the new value. At step 420, the device computes the inverse of r_1 by computing $r_2 = (r_1)^{-1} \bmod \phi(p)$. Because r_1 is not used after this step, its storage space may be used to hold r_2. Note that if $\frac{p-1}{2}$ is prime, then r_2 may also be found by finding $r_2' = r_1^{(p-1)/2-2} \bmod \frac{p-1}{2}$, and using the CRT to find r_2 (mod $p-1$). At step 425, the device updates a_2 by computing $a_2 \leftarrow a_2 r_2 \bmod \phi(p)$. At step 430, the device begins the private key (decryption) process by computing $m' = \gamma^{a_1} \bmod p$. At step 435, the device computes $m = \delta(m')^{a_2} \bmod p$ and returns the message m. If verification is successful, the result equals the original message because:

$$\begin{aligned}(\delta)(m')^{a_2} \bmod p &= \left(m(\alpha^a)^k\right)\left(\gamma^{a_1} \bmod p\right)^{a_2} \bmod p \\ &= \left(m\alpha^{ak}\right)\left(\gamma^{a_1 a_2 \bmod \phi(p)}\right) \bmod p \\ &= \left(m\alpha^{ak}\right)\left((\alpha^k \bmod p)^{-a \bmod \phi(p)}\right) \bmod p \\ &= \left(m\alpha^{ak}\right)\left(\alpha^{-ak}\right) \bmod p \\ &= m\end{aligned}$$

As with the ElGamal public key encryption scheme, the private key for the ElGamal digital signature scheme is a randomly-selected secret a, where $1 \leq a \leq p-2$. The public key is also similar, consisting of a prime p, a generator α, and public parameter y where $y = \alpha^a \bmod p$. To sign a message m, the private key holder chooses or precomputes a random secret integer k (where $1 \leq k \leq p-2$ and k is relatively prime to $p-1$) and its inverse, $k^{-1} \bmod \phi(p)$. Next, the signer computes the signature (r, s), where $r = \alpha^k \bmod p$, $s = \left((k^{-1} \bmod \phi(p))[\mathrm{H}(m) - ar]\right) \bmod \phi(p)$, and $\mathrm{H}(m)$ is the hash of the message. Signature verification is performed using the public key (p, α, y) by verifying that $1 \leq r < p$ and by verifying that $y^r r^s \bmod p = \alpha^{\mathrm{H}(m)} \bmod p$.

To make the ElGamal digital signing process leak-resistant, the token containing the private key maintains three persistent variables, a_k, w, and r. Initially, $a_k = a$ (the private exponent), $w = 1$, and $r = \alpha$. When a message m is to be signed (or during the precomputation before signing), the token generates a random number b and its inverse b^{-1} mod $\phi(p)$, where b is relatively prime to $\phi(p)$ and $0 < b < \phi(p)$. The token then updates a_k, w, and r by computing $a_k \leftarrow (a_k)(b^{-1}) \bmod \phi(p)$, $w \leftarrow (w)(b^{-1}) \bmod \phi(p)$, and $r \leftarrow (r^b) \bmod p$. The signature (r, s) is formed from the updated value of r and s, where $s = (w(\mathrm{H}(m) - a_k r)) \bmod \phi(p)$. Note that a_k, w, and r are not randomized prior to the first operation, but should be randomized before exposure to possible attack, since otherwise the first operation may leak more information than subsequent ones. It is thus recommended that a dummy signature or parameter update with $a_k \leftarrow (a_k)(b^{-1}) \bmod \phi(p)$, $w \leftarrow (w)(b^{-1}) \bmod \phi(p)$, and $r \leftarrow (r^b) \bmod p$ be performed immediately after key generation. Valid signatures produced using the exemplary tamper-resistant ElGamal process may be checked using the normal ElGamal signature verification procedure.

It is also possible to split all or some the ElGamal variables into two halves as part of the leak resistance scheme. In such a variant, a is replaced with a_1 and a_2, w with w_1 and w_2, and r with r_1 and r_2. It is also possible to reorder the operations by performing, for example, the parameter updates as a precomputation step prior to receipt of the enciphered message. Other variations and modifications to the exemplary embodiments described herein will be evident to one skilled in the art.

D. Leak-Resistant DSA

Another commonly used asymmetric cryptographic protocol is the Digital Signature Algorithm (DSA, also known as the Digital Signature Standard, or DSS), which is defined in "Digital Signature Standard (DSS)," Federal Information Processing Standards Publication 186, National Institute of Standards and Technology, May 19, 1994 and described in detail in the Handbook of Applied Cryptography, pages 452 to 454. DSA is widely used for digital signatures. If information about the secret key leaks from a DSA implementation, security can be compromised. Consequently, leak-resistant implementations of DSA would be useful.

In non-leak-proof systems, the private key consists of a secret parameter a, and the public key consists of (p, q, α, y), where p is a large (usually 512 to 1024 bit) prime, q is a 160-bit prime, α is a generator of the cyclic group of order q mod p, and $y = \alpha^a \bmod p$. To sign a message whose hash is $\mathrm{H}(m)$, the signer first generates (or precomputes) a random integer k and its inverse k^{-1} mod q, where $0 < k < q$. The signer then computes the signature (r, s), where $r = (\alpha^k \bmod p) \bmod q$, and $s = (k^{-1} \bmod q)(\mathrm{H}(m) + ar) \bmod q$.

In an exemplary embodiment of a leak-resistant DSA signing process, the token containing the private key maintains two variables in nonvolatile memory, a_k and k, which are initialized with $a_k = a$ and $k = 1$. When a message m is to be signed (or during the precomputation before signing), the token generates a random integer b and its inverse b^{-1} mod q, where $0 < b < q$. The token then updates a_k and k by computing $a_k \leftarrow (a_k b^{-1} \bmod q)(k) \bmod q$, followed by $k \leftarrow b$. The signature (r, s) is formed from the updated values of a_k and k by computing $r = g^k \bmod p$ (which may be reduced mod q), and $s = [(b^{-1}\mathrm{H}(m) \bmod q) + (a_k r) \bmod q] \bmod q$. As indicated, when computing s, $b^{-1}\mathrm{H}(m) \bmod q$ and $(a_k r) \bmod q$ are computed first, then combined mod q. Note that a_k and k should be randomized prior to the first operation, since the first update may leak more information than subsequent updates. It is thus recommended that a dummy signature (or parameter update) be performed immediately after key generation. Valid signatures produced using the leak-resistant DSA process may be checked using the normal DSA signature verification procedure.

IV. Other Algorithms and Applications

Still other cryptographic processes can be made leak-proof or leak-resistant, or may be incorporated into leak-resistant cryptosystems. For example, cryptosystems such as those based on elliptic curves (including elliptic curve analogs of other cryptosystems), secret sharing schemes, anonymous electronic cash protocols, threshold signatures schemes, etc. be made leak resistant using the techniques of the present invention.

Implementation details of the schemes described may be adjusted without materially changing the invention, for example by re-ordering operations, inserting steps, substituting equivalent or similar operations, etc. Also, while new keys are normally generated when a new system is produced, it is often possible to add leak resistance retroactively while maintaining or converting existing private keys.

Leak-resistant designs avoid performing repeated mathematical operations using non-changing (static) secret values, since they are likely to leak out. However, in environments where it is possible to implement a simple function (such as an exclusive OR) that does not leak information, it is possible use this function to implement more complex cryptographic operations.

While the exemplary implementations assume that the leak functions can reveal any information present in the system, designers may often safely use the (weaker) assumption that information not used in a given operation will not be leaked by that operation. Schemes using this weaker assumption may contain a large table of precomputed subkey values, from which a unique or random subset are selected and/or updated for each operation. For example, DES implementations may use indexed permutation lookup tables in which a few table elements are exchanged with each operation.

While leak resistance provides many advantages, the use of leak resistance by itself cannot guarantee good security. For example, leak-resistant cryptosystems are not inherently secure against error attacks, so operations should be verified. (Changes can even be made to the cryptosystem and/or leak resistance operations to detect errors.) Similarly, leak resistance by itself does not prevent attacks that extract the entire state out of a device (e.g., $L=L_{MAX}$). For example, traditional tamper resistance techniques may be required to prevent attackers from staining ROM or EEPROM memory cells and reading the contents under a microscope. Implementers should also be aware of interruption attacks, such as those that involve disconnecting the power or resetting a device during an operation, to ensure that secrets will not be compromised or that a single leaky operation will not be performed repeatedly. (As a countermeasure, devices can increment a counter in nonvolatile memory prior to each operation, and reset or reduce the counter value when the operation completes successfully. If the number of interrupted operations since the last successful update exceeds a threshold value, the device can disable itself.) Other tamper resistance mechanisms and techniques, such as the use of fixed-time and fixed-execution path code or implementations for critical operations, may need to be used in conjunction with leak resistance, particularly for systems with a relatively low self-healing rate (e.g., L_{MAX} is small).

Leak-resistant algorithms, protocols, and devices may be used in virtually any application requiring cryptographic security and secure key management, including without limitation: smartcards, electronic cash, electronic payments, funds transfer, remote access, timestamping, certification, certificate validation, secure e-mail, secure facsimile, telecommunications security (voice and data), computer networks, radio and satellite communications, infrared communications, access control, door locks, wireless keys, biometric devices, automobile ignition locks, copy protection devices, payment systems, systems for controlling the use and payment of copyrighted information, and point of sale terminals.

The foregoing shows that the method and apparatus of the present invention can be implemented using numerous variations and modifications to the exemplary embodiments described herein, as would be known by one skilled in the art. Thus, it is intended that the scope of the present invention be limited only with regard to the claims below.

WHAT IS CLAIMED IS:

1. A method for implementing RSA with the Chinese Remainder Theorem for use in a cryptographic system, with resistance to leakage attacks against said cryptographic system, comprising the steps of:

(a) obtaining a representation of an RSA private key corresponding to an RSA public key, said private key characterized by secret factors p and q;

(b) storing said representation of said private key in a memory;

(c) obtaining a message for use in an RSA cryptographic operation;

(d) computing a first modulus, corresponding to a multiple of p, where the value of said multiple of p and the value of said multiple of p divided by p are both unknown to an attacker of said cryptographic system;

(e) reducing said message modulo said first modulus;

(f) performing modular exponentiation on the result of step (e);

(g) computing a second modulus, corresponding to a multiple of q, where the value of said multiple of q and the value of said multiple of q divided by q are both unknown to an attacker of said cryptographic system;

(h) reducing said message modulo said second modulus;

(i) performing modular exponentiation on the result of step (h);

(j) combining the results of said steps (e) and (h) to produce a result which, if operated on with an RSA public key operation using said RSA public key, yields said message; and

(k) repeating steps (c) through (j) a plurality of times using different values for said multiple of p and for said multiple of q.

2. The method of claim 1 where:

(i) said step (b) includes storing an exponent d_p of said RSA private key in said memory as a plurality of parameters;

(ii) an arithmetic function of at least one of said plurality of parameters is congruent to d_p, modulo (p-1);

(iii) none of said parameters comprising said stored d_p is equal to d_p;

(iv) an exponent used in said step (f) is at least one of said parameters;

(v) at least one of said parameters in said memory changes with said repetitions of said steps (c) through (j).

3. The method of claim 2 where said plurality of parameters includes a first parameter equal to said d_p plus a multiple of phi(p), and also includes a second parameter equal to a multiple of phi(p), where phi denotes Euler's totient function.

4. The method of claim 1 where the value of said multiple of p divided by p is equal to the value of said multiple of q divided by q.

5. The method of claim 1 where said multiple of p and said multiple of q used in said steps (c) through (j) are updated and modified in said memory after said step (b).

6. The method of claim 1 performed in a smart card.

7. The method of claim 1 where at least two of said steps are performed in an order other than (a) through (k)

8. A method for implementing RSA for use in a cryptographic system, with resistance to leakage attacks against said cryptographic system, comprising the steps of:

(a) obtaining an RSA private key corresponding to an RSA public key, said RSA public key having an RSA modulus n;

(b) storing said private key in a memory in a form whereby a secret parameter of said key is stored as an arithmetic combination of phi(x) and a first at least one key masking parameter, where

(i) an operand x in said phi(x) is an exact multiple of at least one factor of said modulus n of said RSA public key; and

(ii) said first key masking parameter is unknown to an attacker of said cryptosystem;

(iii) a representation of said first key masking parameter is stored in said memory;

(iv) phi denotes Euler's totient function;

(c) receiving a message;

(d) deriving an RSA input from said message;

(e) performing modular exponentiation to raise said RSA input to a power dependent on said secret parameter, modulo an RSA modulus stored in said memory, to produce an RSA result such that said RSA result raised to the power of the public exponent of said RSA public key, modulo the modulus of said RSA public key, equals said RSA input;

(f) updating said secret parameter in said memory by:

(i) modifying said first key masking parameter to produce a new key masking parameter, where said modification is performed in a manner such that an attacker with partial useful information about said first key masking parameter has less useful information about said new key masking parameter; and

(ii) using said new key masking parameter to update said secret parameter in said memory;

(g) repeating steps (d) through (f) a plurality of times, where the power used for each of said modular exponentiation steps (e) is different.

9. The method of claim 8 where said operand x in said phi(x) corresponds to said RSA modulus n of said RSA public key.

10. The method of claim 8 where said operand x in said phi(x) corresponds to a prime factor of said RSA modulus n of said RSA public key, and where said modular exponentiation of said step (e) is performed using the Chinese Remainder Theorem.

11. A method for implementing exponential key exchange for use in a cryptographic system, with resistance to leakage attacks against said cryptographic system, comprising the steps of:

(a) obtaining, and storing in a memory, exponential key exchange parameters g and p, and a plurality of secret exponent parameters on which an arithmetic relationship may be computed to produce an exponent x;

(b) using a key update transformation to produce a plurality of updated secret exponent parameters while maintaining said arithmetic relationship thereamong;

(c) receiving a public value y from a party with whom said key exchange is desired;

(d) using said updated secret exponent parameters to perform a cryptographic computation yielding an exponential key exchange result z = y^x mod p;

(e) using said result z to secure an electronic communication with said party; and

(f) performing said steps (b), (c), (d), and (e) in a plurality of transactions.

12. The method of claim 11 where each of said transactions involves a different said party.

13. The method of claim 11 where said arithmetic relationship is such that said exponential key exchange result is a product of certain of said secret exponent parameters, both before and after said step (b).

14. The method of claim 11 where said key update transformation includes choosing a random key update value r; and where said step (b) includes multiplying one of said secret exponent parameters by r and another of said secret exponent parameters by an inverse of r, said multiplication being performed modulo phi(p), where phi is Euler's totient function.

15. The method of claim 11 where said key update transformation includes choosing a random key update value r; and where said step (b) includes adding r to one of said secret exponent parameters and subtracting r from another of said secret exponent parameters.

16. The method of claim 15 where said secret exponent parameters include two values x_1 and x_2 such that x_1+x_2 is congruent to x, modulo phi(p), where phi is Euler's totient function, and where said step of performing said cryptographic computation yielding said exponential key exchange result includes computing $z_1 = y^\wedge x_1 \bmod p$, $z_2 = y^\wedge x_2 \bmod p$, and $z=z_1z_2 \bmod p$.

17. A cryptographic token configured to perform cryptographic operations using a secret key in a secure manner, comprising:
   (a) an interface configured to receive power from a source external to said token;
   (b) a memory containing said secret key;
   (c) a processor:
      (i) configured to receive said power delivered via said interface;
      (ii) configured to perform said processing using said secret key from said memory;
   (d) said token having a power consumption characteristic:
      (i) that is externally measurable; and

(ii) that varies over time in a manner measurably correlated with said cryptographic operations; and

(e) a source of unpredictable information usable in said cryptographic operations to make determination of said secret key infeasible from external measurements of said power consumption characteristic.

18. The cryptographic token of claim 17, in the form of a secure microprocessor.

19. The cryptographic token of claim 17, in the form of a smart card.

20. The cryptographic token of claim 19, wherein said cryptographic operations performed by said smart card enable a holder thereof to decrypt an encrypted communication received via a computer network.

21. The cryptographic token of claim 19, wherein said smart card is configured to store value in an electronic cash scheme.

22. The cryptographic token of claim 21, wherein said cryptographic operations include authenticating that a balance of said stored value has been decreased.

23. The cryptographic token of claim 17, wherein said cryptographic operations include asymmetric private key operations.

24. The cryptographic token of claim 23 wherein said cryptographic operations include exponential key agreement operations.

25. The cryptographic token of claim 23, wherein said cryptographic operations include DSA signing operations.

26. The cryptographic token of claim 23, wherein said cryptographic operations include ElGamal private key operations.

27. The cryptographic token of claim 23, wherein said asymmetric private key operations include RSA private key operations.

28. The cryptographic token of claim 27 wherein said private key operations include Chinese Remainder Theorem operations.

29. The cryptographic token of claim 17, wherein said cryptographic operations include symmetric encryption operations.

30. The cryptographic token of claim 17, wherein said cryptographic operations include symmetric decryption operations.

31. The cryptographic token of claim 17, wherein said cryptographic operations include symmetric authentication operations using said secret key.

32. The cryptographic token of claim 17, wherein said cryptographic operations include authenticating a payment.

33. The cryptographic token of claim 17, wherein said cryptographic operations include securing a broadcast communications signal.

34. The cryptographic token of claim 33, wherein said cryptographic operations include decrypting a satellite broadcast.

35. A method for securely managing and using a private key in a computing environment where information about said private key may leak to attackers, comprising the steps of:

(a) using a first private key, complementary to a public key, to perform first asymmetric cryptographic operation;

(b) reading at least a portion of said first private key from a memory;

(c) transforming said read portion of said first private key to produce a second private key:

(i) said second private key usable to perform a subsequent asymmetric cryptographic operation in a manner that remains complementary to said public key, and

(ii) said transformation enabling said asymmetric cryptographic operations to be performed in a manner such that information leaked during said first asymmetric cryptographic operation does not provide incrementally useful information about said second private key;

(d) obtaining a datum;

(e) using said second private key to perform said subsequent asymmetric cryptographic operation on said datum.

36. The method of claim 35 where said asymmetric cryptographic operation includes a digital signing operation.

37. The method of claim 36 where said signing operation is an RSA operation.

38. The method of claim 36 where said signing operation is an DSA operation.

39. The method of claim 36 where said signing operation is an ElGamal operation.

40. The method of claim 35 where said asymmetric cryptographic operation includes a decryption operation.

41. The method of claim 40 where said decryption operation is an RSA operation.

42. The method of claim 40 where said decryption operation is an ElGamal operation.

43. The method of claim 35 where at least two of said steps are performed in an order different than (a), (b), (c), (d), (e).

44. The method of claim 35 further comprising the step, after at least said step (c), of replacing said private key in said memory with said second private key.

45. The method of claim 35, performed in a smart card.

46. The method of claim 35, further comprising the steps of: prior to at least said step (c), incrementing a counter stored in a nonvolatile memory and verifying that said counter has not exceeded a threshold value; and after at least said step (c) has completed successfully, decreasing a value of said counter.

47. A method for performing cryptographic transactions while protecting a stored cryptographic key against compromise due to leakage attacks, comprising the steps of:

(a) retrieving a stored private cryptographic key stored in a memory, said stored key having been used in a previous cryptographic transaction;

(b) using a first cryptographic function to derive from said stored key an updated key, about which useful information about said stored key obtained through monitoring of leaked information is effectively uncorrelated to said updated key;

(c) replacing said stored key in said memory with said updated key;

(d) using an asymmetric cryptographic function, cryptographically processing a datum with said updated key; and

(e) sending said cryptographically processed datum to an external device having a public key corresponding to said stored key.

48. The method of claim 47 where said stored key includes a first plurality of parameters, and where said updated key includes a second plurality of parameters.

49. The method of claim 48 where no secret value within said first plurality of parameters is included within said second plurality of parameters.

50. The method of claim 49 where said first plurality of parameters is different than said second plurality of parameters, yet a predetermined relationship among said first plurality of parameters is also maintained among said second plurality of parameters.

51. The method of claim 50 where said relationship among said plurality of parameters is an arithmetic function involving at least two of said plurality of parameters.

52. The method of claim 51 where said arithmetic function is the sum of said parameters.

53. The method of claim 51 where said relationship includes a bitwise combination of said parameters.

54. The method of claim 53 where said bitwise combination is an exclusive OR.

55. The method of claim 47 where said step (b) includes using pseudorandomness to derive said updated key.

56. A method for implementing a private key operation for an asymmetric cryptographic system with resistance to leakage attacks against said cryptographic system, comprising the steps of:

(a) encoding a portion of a private key as at least two component parts, such that an arithmetic function of said parts yields said portion;

(b) modifying said component parts to produce updated component parts, but where said arithmetic function of said updated parts still yields said private key portion;

(c) obtaining a message for use in an asymmetric private key cryptographic operation;

(d) separately applying said component parts to said message to produce an intermediate result;

(e) deriving a final result from said intermediate result such that said final result is a valid result of applying said private key to said message; and

(f) repeating steps (b) through (e) a plurality of times.

57. The method of claim 56 where said private key portion includes an exponent, and where said intermediate result represents the result of raising said message to the power of said exponent, modulo a second key portion.

58. The method of claim 57 where said private key operation is configured for use with an RSA cryptosystem.

59. The method of claim 57 where said private key operation is configured for use with an ElGamal cryptosystem.

60. The method of claim 56 where said private key operation is configured for use with a DSA cryptosystem.

61. The method of claim 60 where said private key is represented by secret parameters a_k and k whose product, modulo a predetermined DSA prime q for said private key, yields said private key portion.

62. The method of claim 56 implemented in a smart card.

63. The method of claim 56 where said private key is configured for use with an elliptic curve cryptosystem.

64. A method for performing cryptographic transactions in a cryptographic token while protecting a stored cryptographic key against compromise due to leakage attacks, including the steps of:

(a) retrieving said stored key from a memory;

(b) cryptographically processing said key, to derive an updated key, by executing a cryptographic update function that:

(i) prevents partial information about said stored key from revealing useful information about said updated key, and

(ii) also prevents partial information about said updated key from revealing useful information about said stored key;

(c) replacing said stored key in said memory with said updated key;

(d) performing a cryptographic operation using said updated key; and

(e) repeating steps (a) through (d) a plurality of times.

65. The method of claim 64 where said cryptographic update function of said step (b) includes a one-way hash operation.

66. The method of claim 64 where said cryptographic operation of said step (d) is a symmetric cryptographic operation; and comprising the further step of sending a result of said cryptographic operation to a party capable of rederiving said updated key.

67. The method of claim 64 further comprising the step, prior to said step (a), of receiving from a second party a symmetric authentication code and a parameter; and said where said step (b) includes iterating a cryptographic transformation a number of times determined from said parameter; and where said step (d) includes performing a symmetric message authentication code verification operation.

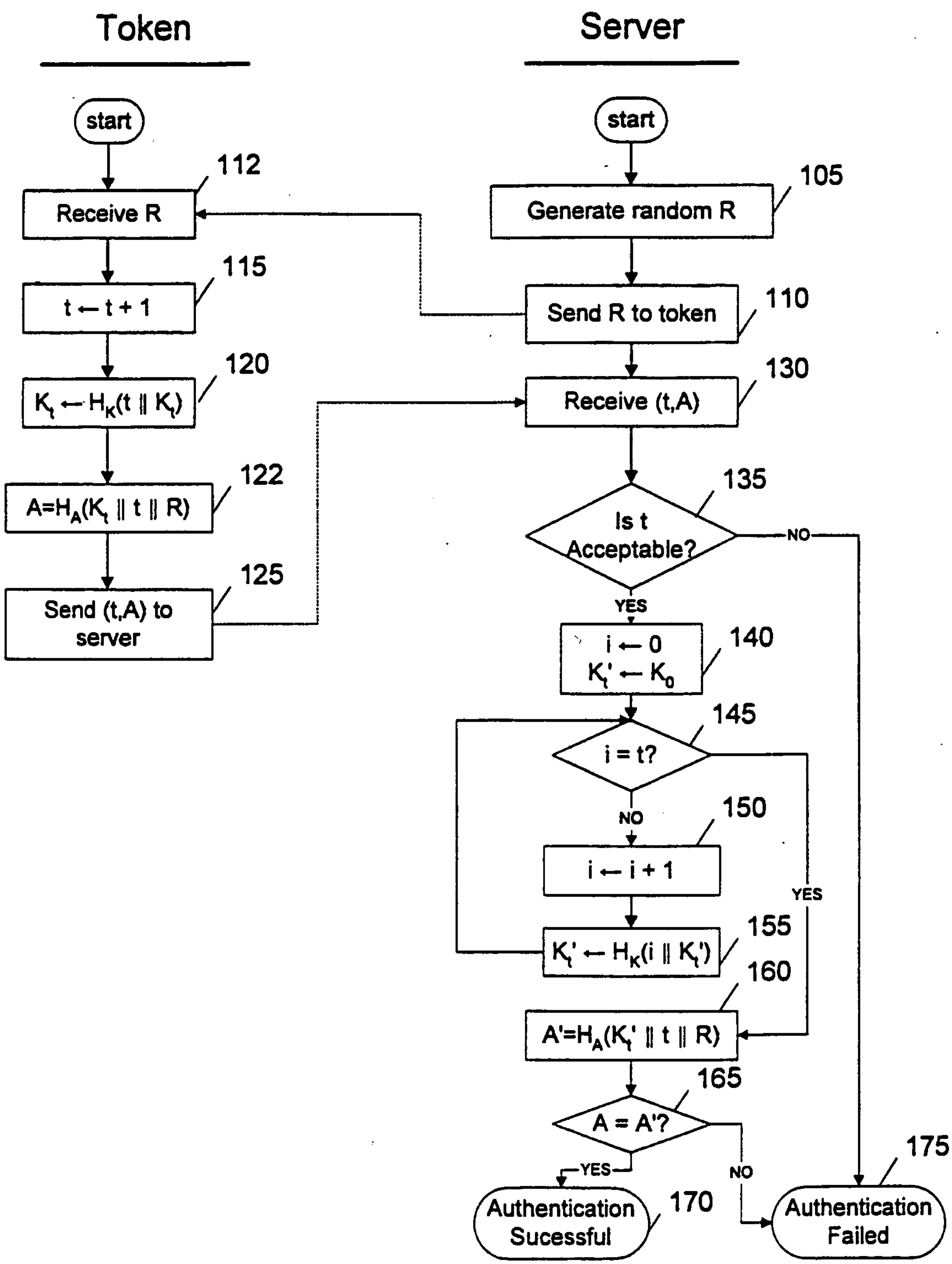
68. he method of claim 66 where said step (d) of performing said cryptographic operation includes using said updated key to encrypt a datum.

69. The method of claim 66 where said updated key contains unpredictable information such that said updated key is not stored in its entirety anywhere outside of said cryptographic token; and where the result of said step (d) is independent of said unpredictable information.

70. The method of claim 64 where said step (c) of replacing said stored key includes:
(i) explicitly erasing a region of said memory containing said stored key; and
(ii) storing said updated key in said region of memory.

71. The method of claim 64 performed within a smart card.

FIG. 1



SUBSTITUTE SHEET (RULE 26)

FIG. 2

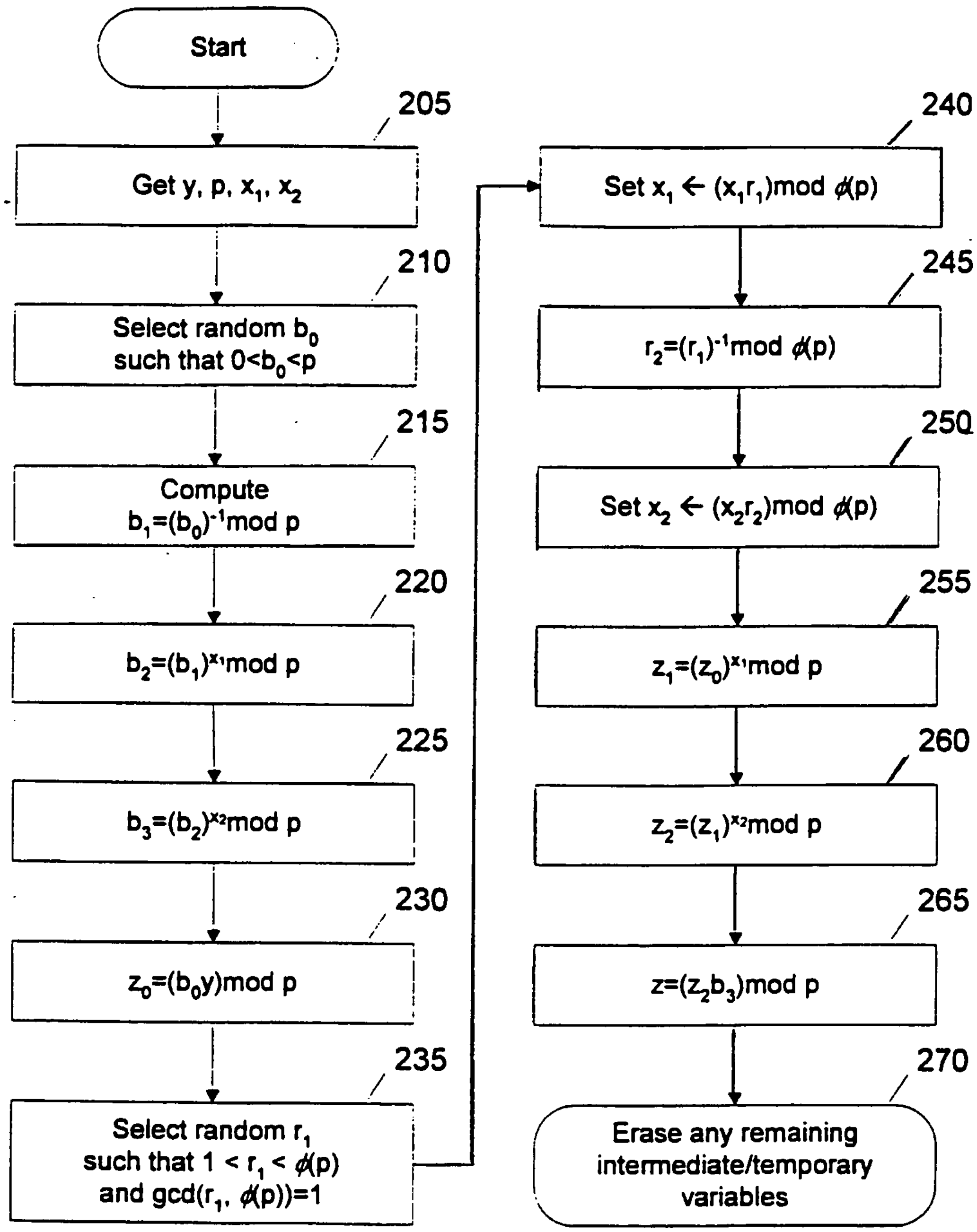

Start
205
Get y, p, x_1, x_2
210
Select random b_0 such that $0<b_0<p$
215
Compute $b_1=(b_0)^{-1} \bmod p$
220
$b_2=(b_1)^{x_1} \bmod p$
225
$b_3=(b_2)^{x_2} \bmod p$
230
$z_0=(b_0 y) \bmod p$
235
Select random r_1 such that $1 < r_1 < \phi(p)$ and $\gcd(r_1, \phi(p))=1$
240
Set $x_1 \leftarrow (x_1 r_1) \bmod \phi(p)$
245
$r_2=(r_1)^{-1} \bmod \phi(p)$
250
Set $x_2 \leftarrow (x_2 r_2) \bmod \phi(p)$
255
$z_1=(z_0)^{x_1} \bmod p$
260
$z_2=(z_1)^{x_2} \bmod p$
265
$z=(z_2 b_3) \bmod p$
270
Erase any remaining intermediate/temporary variables

FIG. 3

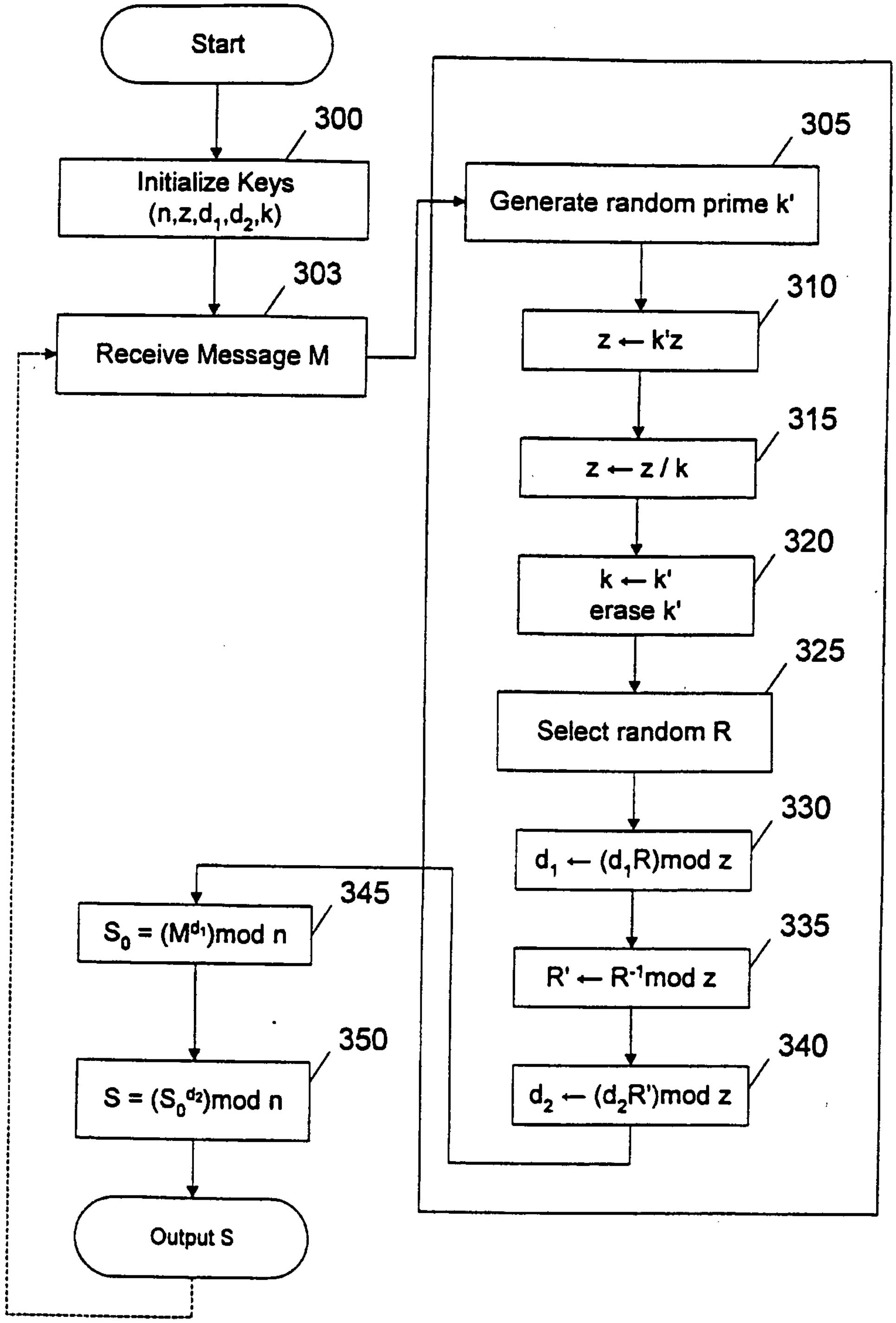
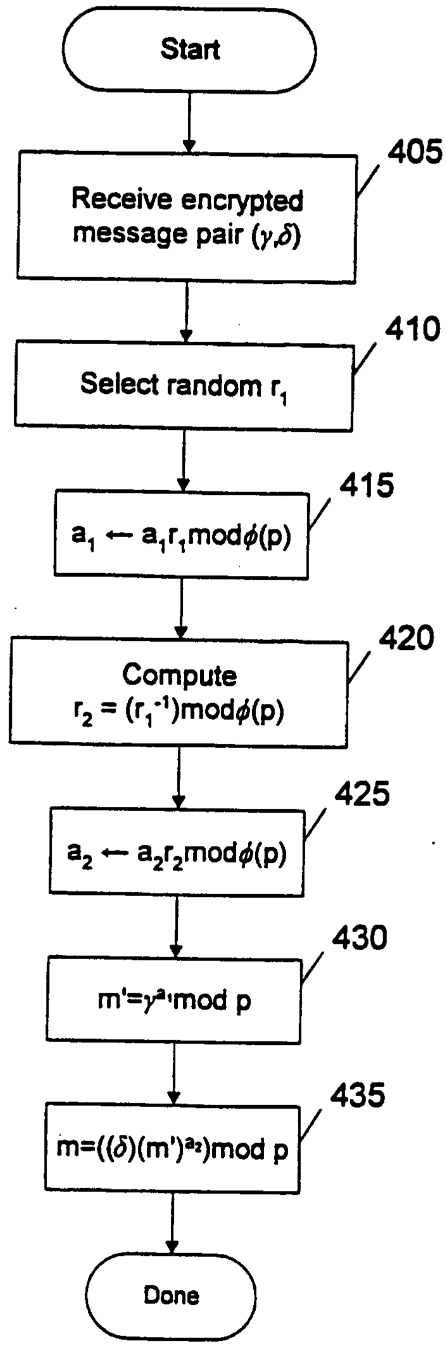

Start
300
Initialize Keys
(n,z,d_1,d_2,k)
303
Receive Message M
305
Generate random prime k'
310
$z \leftarrow k'z$
315
$z \leftarrow z / k$
320
$k \leftarrow k'$
erase k'
325
Select random R
330
$d_1 \leftarrow (d_1R) \bmod z$
335
$R' \leftarrow R^{-1} \bmod z$
340
$d_2 \leftarrow (d_2R') \bmod z$
345
$S_0 = (M^{d_1}) \bmod n$
350
$S = (S_0^{d_2}) \bmod n$
Output S

FIG. 4


Start
405
Receive encrypted message pair (γ,δ)
410
Select random r_1
415
$a_1 \leftarrow a_1 r_1 \bmod \phi(p)$
420
Compute
$r_2 = (r_1^{-1}) \bmod \phi(p)$
425
$a_2 \leftarrow a_2 r_2 \bmod \phi(p)$
430
$m' = \gamma^{a_1} \bmod p$
435
$m = ((\delta)(m')^{a_2}) \bmod p$
Done

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/27896

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :HO4 L 9/30
US CL :380/30,49
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. : 380/30,49

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)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 4,799,258 A (DAVIES et al) 17 January 1989, abstract, col.4, lines 43-50, col.7,lines 15-33, col.8,lines 12-19	17-23,25- 45
Y	US 5,546,463 A (CAPUTO et al) 13 August 1996, abstract, col.2, lines, 60-65, col.5, lines 39-50,53-58, col.6, lines 7-12	17-23,25-45
A,P	US 5,848,159 A (COLLINS et al.) 08 December 1998, abstract, col.1, lines 56-67, col.4, lines 33-44, col.5, lines 52-67, col.6, lines 24-30	1-16,46-71

☐ Further documents are listed in the continuation of Box C. ☐ See patent family 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 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	Date of mailing of the international search report
30 MARCH 1999	06 MAY 1999

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	GAIL HAYES Eugenia Zagen
Facsimile No. (703) 305-0040	Telephone No. (703) 305-9711

Form PCT/ISA/210 (second sheet)(July 1992) ★

B. FIELDS SEARCHED
Electronic data bases consulted (Name of data base and where practicable terms used):

APS
search terms:token,smart card,tamper proof, tamper resistant,leak-resistant,RSA, public key, private key, chinese remainder theorem, diffie hellman,dsa,des

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04N 7/167, G06F 1/00

A1

(11) International Publication Number: **WO 99/48296**

(43) International Publication Date: 23 September 1999 (23.09.99)

(21) International Application Number: PCT/US99/05734

(22) International Filing Date: 16 March 1999 (16.03.99)

(30) Priority Data:
60/078,053 16 March 1998 (16.03.98) US

(71) Applicant: INTERTRUST TECHNOLOGIES CORPORATION [US/US]; 460 Oakmead Parkway, Sunnyvale, CA 94086 (US).

(72) Inventors: SHAMOON, Talal, G.; 533 Bryant Street #5, Palo Alto, CA 94301 (US). HILL, Ralph, D.; 224 Dover Street, Los Gatos, CA 94032 (US). RADCLIFFE, Chris, D.; 3654 Farm Hill Boulevard, Redwood City, CA 94061 (US). HWA, John, P.; 503 Lower Vinters Circle, Fremont, CA 94539 (US).

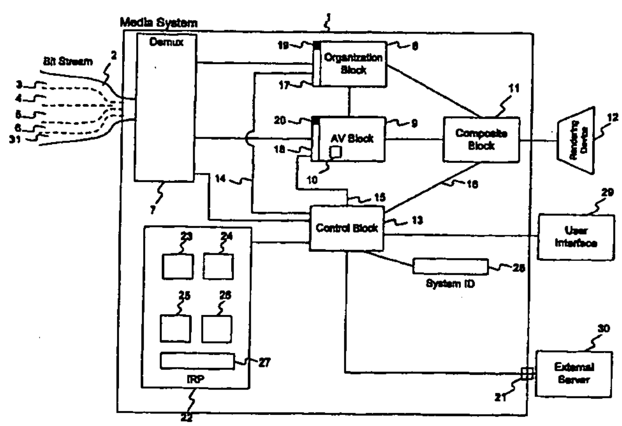
(74) Agents: GARRETT, Arthur, S. et al.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 1300 I Street, Washington, DC 20005–3315 (US).

(81) Designated States: CA, CN, JP, KR, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published

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.

(54) Title: METHODS AND APPARATUS FOR CONTINUOUS CONTROL AND PROTECTION OF MEDIA CONTENT



(57) Abstract

A novel method and apparatus for protection of streamed media content is disclosed. The apparatus includes control means for governance of content streams or objects, decryption means for decrypting content streams or objects under control of the control means, and feedback means for tracking actual use of content streams or objects. The control means may operate in accordance with rules received as part of the streamed content, or through a side–band channel. The rules may specify allowed uses of the content, including whether or not the content can be copied or transferred, and whether and under what circumstances received content may be "checked out" of one device and used in a second device. The rules may also include or specify budgets, and a requirement that audit information be collected and/or transmitted to an external server. The apparatus may include a media player designed to call plugins to assist in rendering content. A "trust plugin" and its use are disclosed so that a media player designed for use with unprotected content may render protected content without the necessity of requiring any changes to the media player. The streamed content may be in a number of different formats, including MPEG–4, MP3, and the RMFF format.

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		

METHODS AND APPARATUS FOR CONTINUOUS CONTROL AND PROTECTION OF MEDIA CONTENT

FIELD OF THE INVENTION

This invention relates generally to computer and/or electronic security. More particularly, this invention relates to systems and methods for protection of information in streamed format.

BACKGROUND

Streaming digital media consists generally of sequences of digital information received in a "stream" of packets, and designed to be displayed or rendered. Examples include streamed audio content, streamed video, etc.

Digital media streams are becoming an increasingly significant means of content delivery, and form the basis for several adopted, proposed or de facto standards. The acceptance of this format, however, has been retarded by the ease with which digital media streams can be copied and improperly disseminated, and the consequent reluctance of content owners to allow significant properties to be distributed through streaming digital means. For this reason, there is a need for a methodology by which digital media streams can be protected.

SUMMARY OF THE INVENTION

Consistent with the invention, this specification describes a new architecture for protection of information provided in streamed format. This architecture is described in the context of a generic system which resembles a system to render content encoded pursuant to the MPEG-4 specification (ISO/IEC 14496.1), though with certain modifications, and with the proviso that the described system may differ from the MPEG-4 standard in certain respects. A variety of different embodiments is described, including an MPEG-4 embodiment and a system designed to render content encoded pursuant to the MP3 specification (ISO/IEC TR 11172).

According to aspects of the invention, this architecture involves system design aspects and information format aspects. System design aspects include the incorporation of content protection functionality, control functionality, and feedback enabling control functionality to monitor the activities of the system. Information format aspects include the incorporation of rule/control information into information streams, and the protection of content through mechanisms such as encryption and watermarking.

Systems and methods consistent with the present invention perform content protection and digital rights management. A streaming media player consistent with the present invention includes a port designed to accept a digital bit stream. The digital bit stream includes content, which is encrypted at least in part, and a secure container including control information designed to control use of the content, including at least one key suitable for decryption of at least a portion of the content. The media player also includes a control arrangement including a means for opening secure containers and extracting cryptographic keys, and means for decrypting the encrypted portion of the content.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, serve to explain the advantages and principles of the invention. In the drawings,

FIG. 1 shows a generic system consistent with the present invention;

FIG. 2 shows an exemplary Header 201 consistent with the present invention;

FIG. 3 shows a general encoding format consistent with the present invention;

FIG. 4 illustrates one manner for storing a representation of a work consistent with the present invention;

FIG. 5 shows an example of a control message format;

FIG. 6 is a flow diagram illustrating one embodiment of the steps which take place using the functional blocks of FIG. 1;

FIG. 7 illustrates a form wherein the control messages may be stored in Control Block 13;

FIG. 8 shows MPEG-4 System 801 consistent with the present invention;

FIG. 9 shows an example of a message format;

FIG. 10 illustrates an IPMP table consistent with the present invention;

FIG. 11 illustrates a system consistent with the present invention;

FIG. 12 illustrates one embodiment of the DigiBox format;

FIG. 13 shows an example of a Real Networks file format (RMFF);

FIG. 14 shows an RNPFF format consistent with the present invention;

FIG. 15 illustrates the flow of changes to data in the Real Networks file format in an architecture consistent with the present invention;

FIG. 16 illustrates a standard Real Networks architecture;

FIG. 17 shows an exemplary architecture in which a trust plugin operates within the overall Real Networks architecture;

FIG. 18 shows a bit stream format consistent with the principles of the present invention;

FIG. 19 shows one embodiment of protection applied to the MP3 format;

FIG. 20 illustrates one embodiment of an MP3 player designed to process and render protected content;

FIG. 21 illustrates the flow of data in one embodiment in which a protected MPEG-4 file may be created consistent with the present invnetion;

FIG. 22 illustrates the flow of data in one embodiment in which control may be incorporated into an existing MPEG-4 stream consistent with the present invention;

FIG. 23 shows a system consistent with the principles of the present invention;

FIG. 24 shows a system consistent with the principles of the present invention;

FIG. 25 shows an example of an aggregate stream consistent with the present invention;

FIG. 26 illustrates a Header CMPO 2601 consistent with the present invention;

FIG. 27 shows exemplary Content Management Protection Objects consistent with the principles of the present invention; and

FIG. 28 shows an example of a CMPO Data Structure 2801 consistent with the present invention.

DETAILED DESCRIPTION

Reference will now be made in detail to implementations consistent with the principles of the present invention as illustrated in the accompanying drawings.

The following U.S. patents and applications, each of which is assigned to the assignee of the current application, are hereby incorporated in their entirety by reference: Ginter, et al., "Systems and Methods for Secure Transaction Management and Electronic Rights Protection," U.S. Patent Application Serial No. 08/964,333, filed on November 4, 1997 ("Ginter '333"); Ginter, et al., "Trusted Infrastructure Support Systems, Methods and Techniques for Secure electronic commerce, Electronic Transactions, Commerce Process Control Automation, Distributed Computing, and Rights Management, " U.S. Patent Application Serial No. 08/699,712, filed on August 12, 1996 ("Ginter '712") ; Van Wie, et al, "Steganographic Techniques for Securely Delivering Electronic Digital Rights Management Information Over Insecure Communications Channels, U.S. Patent Application Serial No. 08/689,606, filed on August 12, 1996 ("Van Wie") ; Ginter, et. al "Software Tamper Resistance and Secure Communication," U.S. Patent Application Serial No. 08/706,206, filed on August 30, 1996 ("Ginter, '206"); Shear, et al, "Cryptographic Methods, Apparatus & Systems for Storage Media Electronic Rights Management in

Closed & Connected Appliances, " U.S. Patent Application Serial No. 08/848,077, filed on May 15, 1997 ("Shear"); Collberg et al, "Obfuscation Techniques for Enhancing Software Security, " U.S. Patent Application Serial No. 09/095,346, filed on June 9, 1998 ("Collberg"); Shear, "Database Usage Metering and Protection System and Method," U.S. Patent No. 4,827,508, issued on May 2, 1989 ("Shear Patent").

FIG. 1 illustrates Media System 1, which is capable of accepting, decoding, and rendering streamed multimedia content. This is a generic system, though it includes elements based on the MPEG-4 specification. Media System 1 may include software modules, hardware (including integrated circuits) or a combination. In one embodiment, Media System 1 may include a Protected Processing Environment (PPE) as described in the Ginter '333 application.

In FIG. 1, Bit Stream 2 represents input information received by System 1. Bit Stream 2 may be received through a connection to an external network (e.g., an Internet connection, a cable hookup, radio transmission from a satellite broadcaster, etc.), or may be received from a portable memory device, such as a DVD player.

Bit Stream 2 is made up of a group of related streams of information, including Organization Stream 3, Audio Stream 4, Video Stream 5, Control Stream 6, and Info Stream 31. Each of these streams is encoded into the overall Bit Stream 2. Each of these represents a category of streams, so that, for example, Video Stream 5 may be made up of a number of separate Video Streams.

These streams correspond generally to streams described in the MPEG-4 format as follows:

Organization Stream 3 corresponds generally to the BIFS stream and the OD ("Object Descriptor") stream.

Audio Stream 4 and Video Stream 5 correspond generally to the Audio and Video streams.

Control Stream 6 corresponds generally to the IPMP stream.

Audio Stream 4 includes compressed (and possibly encrypted) digital audio information. This information is used to create the sound rendered and output by Media System 1. Audio Stream 1 may represent multiple audio streams. These multiple streams may act together to make up the audio output, or may represent alternative audio outputs.

Video Stream 5 includes compressed (and possibly encrypted) digital video information. This information is used to create the images and video rendered and output by Media System 1. Video Stream 5 may represent multiple video streams. These multiple streams may act together to make up the video output, or may represent alternative video outputs.

Organization Stream 3 includes organizational information and metadata related to the work to be rendered. This information may include a tree or other organizational device which groups audio and video streams into objects. This information may also include metadata associated with the entire work, the objects, or the individual streams.

Control Stream 6 includes control information, divided generally into header information and messages. The header information includes an identifier for each discrete message. The content of the messages, which will be described further below, may include cryptographic keys and rules governing the use of content.

Info Stream 31 carries additional information associated with the content in other components of Bit Stream 2, including but not limited to graphics representing cover art, text for lyrics, coded sheet music or other notation, independent advertising content, concert information, fan club information, and so forth. Info Stream 31 can also carry system management and control information and/or components, such as updates to software or firmware in Media System 1, algorithm implementations for content-specific functions such as watermarking, etc.

Each of these streams is made up of packets of information. In one exemplary embodiment, each packet is 32 bytes in length. Since a single communications channel (e.g., a cable, a bus, an infrared or radio connection) contains packets from each of the streams, packets need to be identified as belonging to a particular stream. In a preferred embodiment, this is done by including a header which identifies a particular stream and specifies the number of following packets which are part of that stream. In another embodiment, each packet may include individual stream information.

Exemplary Header 201 is shown in FIG. 2. This header may generally be used for the Organization, Audio and Video Streams. A header for the Control Stream is described below. Header 201 includes Field 202, which includes a bit pattern identifying Header 201 as a header. Field 203 identifies the particular type of stream (e.g., Audio Stream, Organization Stream, Control Stream, etc.) Field 204 contains an Elementary Stream Identifier (ES_ID), which is used to identify the particular stream, and may be used in cases where multiple streams of a particular stream type may be encountered at the same time. Field 207 contains a time stamp, which is used by the system to synchronize the various streams, including rendering of the streams. Composite Block 11 may, for example, keep track of the elapsed time from the commencement of rendering. Time Stamp 207 may be used by Composite Block 11 to determine when each object is supposed to be rendered. Time Stamp 207 may therefore specify an elapsed time from commencement of rendering, and Composite Block 11 may use that elapsed time to determine when to render the associated object.

Field 205 contains a Governance Indicator. Field 206 identifies the number of following packets which are part of the identified stream. In each case, the relevant information is encoded in a binary format. For example, Field 202 might include an arbitrary sequence of bits which is recognized as indicating a header, and Field 203 might include two bits, thereby allowing encoding of four different stream types.

Returning to FIG. 1, System 1 includes Demux 7, which accepts as input Bit Stream 2 and routes individual streams (sometimes referred to as Elementary Streams or "ESs") to appropriate functional blocks of the system.

Bit Stream 2 may be encoded in the format illustrated in FIG. 3. In this figure, Header 301 is encountered in the bit stream, with Packet 302 following, and so on through Packet 308.

When Demux 7 encounters Header 301, Demux 7 identifies Header 301 as a header and uses the header information to identify Packets 302-305 as organization stream packets. Demux 7 uses this information to route these packets to Organization Block 8. Demux 7 handles Header 306 in a similar manner, using the contained information to route Packets 307 and 308 to AV ("Audio Video") Block 9.

AV Block 9 includes Decompressor 10, which accepts Elementary Streams from Audio Stream 4 and Video Stream 5 and decompresses those streams. As decompressed, the stream information is placed in a format which allows it to be manipulated and output (through a video display, speakers, etc.). If multiple streams exist (e.g., two video streams each describing an aspect of a video sequence), AV Block 9 uses the ES_ID to assign each packet to the appropriate stream.

Organization Block 8 stores pointer information identifying particular audio streams and video streams contained in a particular object, as well as metadata information describing, for example, where the object is located, when it is to be displayed (e.g., the time stamp associated with the object), and its relationship to other objects (e.g., is one video object in front of or behind another video object). This organization may be maintained hierarchically, with individual streams represented at the lowest level, groupings of streams into objects at a higher level, complete scenes at a still higher level, and the entire work at the highest level.

FIG. 4 illustrates one manner in which Organization Block 8 may store a representation of a work. In this Figure, Tree 401 represents an entire audiovisual work. Branch 402 represents a high-level organization of the work. This may include, for example, all of the video or possibly the audio and video associated with a particular scene.

Sub-Branch 403 represents a group of related video objects. Each such object may include an entire screen, or an individual entity within the screen. For example, Sub-Branch 403 may represent a background which does not change significantly from one shot to the next. If the video is moving between two points of reference (e.g., a conversation, with the camera point of view changing from one face to the other), Sub-Branch 404 could represent a second background, used in the second point of view.

Nodes 405 and 406 may represent particular video objects contained within the related group. Node 405 could, for example, represent a distant mountain range, while Node 406 represents a tree immediately behind one of the characters.

Each of the nodes specifies or contains a particular ES_ID, representing the stream containing the information used by that node. Node 405, for example, contains ES_ID 407, which identifies a particular video stream which contains compressed (and possibly encrypted) digital information representing the mountain range.

Composite Block 11 accepts input from Organization Block 8 and from AV Block 9. Composite Block 11 uses the input from Organization Block 8 to determine which specific audiovisual elements will be needed at any given time, and to determine the organization and relationship of those elements. Composite Block 11 accepts decompressed audiovisual objects from AV Block 9, and organizes those objects as specified by information from Organization Block 8. Composite Block 11 then passes the organized information to Rendering Device 12, which might be a television screen, stereo speakers, etc.

Control Block 13 stores control messages which may be received through Control Stream 6 and/or may be watermarked into or steganographically encoded in other streams, including Audio Stream 4 and Video Stream 5. One control message format is illustrated by FIG. 5, which shows Control Message 501. Control Message 501 is made up of Header 502 and Message 503. Header 502 consists of Field 508, which includes a bit pattern identifying the following information as a header; Stream Type Field 509, which identifies this as a header for the organization stream; ID Field 504, which identifies this particular control message; Pointer Field 505, which identifies those ESs which are controlled by this message; Time Stamp Field 507, which identifies the particular portion of the stream which is controlled by this control message (this may indicate that the entirety of the stream is controlled); and Length Field 506, which specifies the length (in bytes) of Message 503. Message 503 may include packets following Header 502, using the general format shown in FIG. 3. In the example shown, Control Message 501 carries the unique ID 111000, encoded in ID Field 504. This control message controls ESs 14 and 95, as indicated by Pointer Field 505. The associated Message contains 1,024 bytes, as indicated by Length Field 506.

In an alternate embodiment, the association of control to content may be made in Organization Block 8, which may store a pointer to particular control messages along with the metadata associated with streams, objects, etc. This may be disadvantageous, however, in that it may be desirable to protect this association from discovery or tampering by users. Since Control Block 13 will generally have to be protected in any event, storing the association in this block may make protection of Organization Block 8 less necessary.

Control Block 13 implements control over System 1 through Control Lines 14, 15 and 16, which control aspects of Organization Block 8, AV Block 9 and Composite Block 11, respectively. Each of these Control Lines may allow two-way communication.

Control Lines 14 and 15 are shown as communicating with AV Block Stream Flow Controller 18 and with Organization Block Stream Flow Controller 17. These Stream Flow Controllers contain functionality controlled by Control Block 13. In the embodiment illustrated, the Stream Flow Controllers are shown as the first stage in a two-stage pipeline, with information being processed by the Stream Flow Controller and then passed on to the associated functional block. This allows isolation of the control functionality from the content manipulation and display functionality of the system, and allows control to be added in without altering the underlying functionality of the blocks. In an alternate embodiment, the Stream Flow Controllers might be integrated directly into the associated functional blocks.

Stream Flow Controllers 17 and 18 contain Cryptographic Engines 19 and 20, respectively. These Cryptographic Engines operate under control of Control Block 13 to decrypt and/or cryptographically validate (e.g., perform secure hashing, message authentication code, and/or digital signature functions) the encrypted packet streams received from Demux 7. Decryption and validation may be selective or optional according to the protection requirements for the stream.

Cryptographic Engines 19 and 20 may be relatively complex, and may, for example, include a validation calculator that performs cryptographic hashing, message authentication code calculation, and/or other cryptographic validation processes. In addition, as is described further below, additional types of governance-related processing may also be used. In one alternative embodiment, a single Stream Flow Controller may be used for both Organization Stream 3 and Audio/Video Streams 4-5. This may reduce the cost of and space used by System 1. These reductions may be significant, since System 1 may contain multiple AV Blocks, each handling a separate Audio or Video Stream in parallel. This alternative may, however, impose a latency overhead which may be unacceptable in a real-time system.

If the Stream Flow Controllers are concentrated in a single block, they may be incorporated directly into Demux 7, which may handle governance processing prior to routing streams to the functional blocks. Such an embodiment would allow for governed decryption or validation of the entirety of Bit Stream 2, which could occur prior to the routing of streams to individual functional blocks. Encryption of the entirety of Bit Stream 2 (as opposed to individual encryption of individual ESs) might be difficult or impossible without incorporating stream controller functionality into Demux 7, since Demux 7 might otherwise have no ability to detect or read the header information necessary to route streams to functional blocks (that header information presumably being encrypted).

As is noted above, each of the individual streams contained in Bit Stream 2 may be individually encrypted. An encrypted stream may be identified by a particular indicator in the header of the stream, shown in FIG. 2 as Governance Indicator 205.

When a header is passed by Demux 7 to the appropriate functional block, the stream flow controller associated with that block reads the header and determines whether the following packets are encrypted or otherwise subject to governance. If the header indicates that no governance is used, the stream flow controller passes the header and the packets through to the functional blocks with no alteration. Governance Indicator 205 may be designed so that conventionally encoded content (e.g., unprotected MPEG-4 content) is recognized as having no Governance Indicator and therefore passed through for normal processing.

If a stream flow controller detects a set governance indicator, it passes the ES_ID associated with that stream and the time stamp associated with the current packets to Control Block 13 along Control Line 14 or 15. Control Block 13 then uses the ES_ID and time stamp information to identify which control message(s) are associated with that ES. Associated messages are then invoked and possibly processed, as may be used for governance purposes.

A simple governance case is illustrated by FIG. 6, which shows steps which take place using the functional blocks of FIG. 1. In Step 601, Demux 7 encounters a header, and determines that the header is part of the AV stream. In Step 602, Demux 7 passes the header to AV Stream Controller 18. In Step 603, AV Stream Controller 18 reads the header and determines that the governance indicator is set, thereby triggering further processing along Path 604. In Step 605, AV Stream Controller 18 obtains the ES_ID and time stamp from the header and transmits these to Control Block 13, along Control Line 15. In Step 606, Control Block 13 looks up the ES_ID and determines that the ES_ID is associated with a particular control message. In Step 611, Control Block 13 uses the time stamp information to choose among control messages, if there is more than one control message associated with a particular ES. In Step 607, Control Block 13 accesses the appropriate control message, and obtains a cryptographic key or keys for decryption and/or validation. In Step 608, Control Block 13 passes the cryptographic key(s) along Control Line 15 to AV Stream Controller 18. In Step 609, AV Stream Controller 18 uses the cryptographic key as an input to Cryptographic Engine 20, which decrypts and/or validates the packets following the header as those packets are received from Demux 7. In Step 610, the decrypted packets are then passed to AV Block 9, which decompresses and processes them in a conventional manner.

Time stamp information may be useful when it is desirable to change the control message applicable to a particular ES. For example, it may be useful to encode different portions of a stream with different keys, so that an attacker breaking one key (or even a number of keys) will not be able to use the content. This can be done by associating a number of control messages with the same stream, with each control message being valid for a particular period. The time stamp information would then be used to choose which control message (and key) to use at a particular time. Alternatively, one control message may be used, but with updated information being passed in through the Control Stream, the updates consisting of a new time stamp and a new key.

In an alternative embodiment, Control Block 13 may proactively send the appropriate keys to the appropriate stream flow controller by using time stamp information to determine when a key will be will be needed. This may reduce overall latency.

Control Line 16 from FIG. 1 comes into play once information has been passed from Organization Block 8 and AV Block 9 to Composite Block 11, and the finished work is prepared for rendering through Rendering Device 12. When Composite Block 11 sends an object to Rendering Device 11, Composite Block 11 sends a start message to Control Block 13. This message identifies the object (including any associated ES_IDs), and specifies the start time of the display (or other rendering) of that object. When an object is no longer being rendered, Composite Block 11 sends an end message to Control Block 13, specifying that rendering of the object has ended, and the time at which the ending occurred. Multiple copies of a particular object may be rendered at the same time. For this reason, start and stop messages sent by Composite Block 11 may include an assigned instance ID, which specifies which instance of an object is being rendered.

Control Block 13 may store information relating to start and stop times of particular objects, and/or may pass this information to external devices (e.g., External Server 30) through Port 21. This information allows Control Block 13 to keep track not only of which objects have been decrypted, but of which objects have actually been used. This may be used, since System 1 may decrypt, validate, and/or decompress many more objects than are actually used. Control Block 13 can also determine the length of use of objects, and can determine which objects have been used together. Information of this type may be used for sophisticated billing and auditing systems, which are described further below.

Control Line 16 may also be used to control the operation of Composite Block 11. In particular, Control Block 13 may store information specifying when rendering of a particular object is valid, and may keep track of the number of times an object has been rendered. If Control Block 13 determines that an object is being rendered illegally (i.e., in violation of rules controlling rendering), Control Block 13 may terminate operation of Composite Block 11, or may force erasure of the illegal object.

In an alternate embodiment, the level of control provided by Control Line 16 may at least in part be provided without requiring the presence of that line. Instead, Control Block 13 may store a hash of the organization information currently valid for Organization Block 8. This hash may be received through Control Stream 6, or, alternatively, may be generated by Control Block 13 based on the information contained in Organization Block 8.

Control Block 13 may periodically create a hash of the information currently resident in Organization Block 8, and compare that to the stored hash. A difference may indicate that an unauthorized alteration has been made to the information in Organization Block 8, thereby potentially allowing a user to render information in a manner violative of the rules associated with that information. In such an event, Control Block 13 may take appropriate action, including deleting the information currently resident in Organization Block 8.

If System 1 is designed so that Organization Block 8 controls the use of content by Composite Block 11, so that content cannot be rendered except as is specified by the organization information, Control Block 13 may be able to control rendering of information through verifying that the current Organization Block contents match the hash which has been received by Control Block 13, thereby eliminating at least one reason for the presence of Control Line 16.

Control Block 13 may also be responsible for securely validating the origin, integrity, authenticity, or other properties of received content, through cryptographic validation means such as secure hashing, message authentication codes, and/or digital signatures.

System 1 may also include an Inter-Rights Point, indicated as IRP 22. IRP 22 is a protected processing environment (e.g., a PPE) in which rules/controls may be processed, and which may store sensitive information, such as cryptographic keys. IRP 22 may be incorporated within Control Block 13, or may be a separate module. As is illustrated, IRP 22 may include CPU 23 (which can be any type of processing unit), Cryptographic Engine 24, Random Number Generator 25, Real Time Clock 26, and Secure Memory 27. In particular embodiments, some of these elements may be omitted, and additional functionality may be included.

Governance Rules

Control messages stored by Control Block 13 may be very complex. FIG. 7 illustrates the form in which the control messages may be stored in Control Block 13, consisting of Array 717. Column 701 consists of the address at which the control messages are stored. Column 702 consists of the identifier for each control message. This function may be combined with that of Column 701, by using the location information of Column 701 as the identifier, or by storing the message in a location which corresponds to the identifier. Column 703 consists of the ES_IDs for each stream controlled by the control message. Column 704 consists of the message itself. Thus, the control message stored at location 1 has the ID 15, and controls stream 903.

In a simple case, the message may include a cryptographic key, used to decrypt the content associated with the stream(s) controlled by the message. This is illustrated by Cryptographic Key 705 from FIG. 7. Cryptographic keys and/or validation values may also be included to permit cryptographic validation of the integrity or origin of the stream.

In a more complex case, the message may include one or more rules designed to govern access to or use of governed content. Rules may fall into a number of categories.

Rules may require that a particular aspect of System 1, or a user of System 1, be verified prior to decryption or use of the governed content. For example, System 1 may include System ID 28, which stores a unique identifier for the system. A particular rule contained in a control message may specify that a particular stream can only be decrypted on a system in which System ID 28 contains a particular value. This is illustrated at row 2 in FIG. 7, in which the message is shown as consisting of a rule and commands. The rule may be implicit, and therefore may not be stored explicitly in the table (e.g. the table may store only the rule, the rule - specific functions (commands) invoked by the rule, or only the functions).

In this case, when Stream Controller 18 encounters a Header for stream 2031 containing a set governance indicator, Stream Controller 18 passes the associated ES_ID (2031) to Control Block 13. Control Block 13 then uses the ES_ID to identify Control Message 20 which governs stream 2031. Control Message 20 includes Rule 706, which includes (or invokes) Commands 707, and an Authorized System ID 708. Authorized System ID 708 may have been received by System 1, either as part of Control Message 20, or as part of another control message (e.g., Control Message 9), which Control Message 20 could then reference in order to obtain access to the Authorized System ID. Such a case might exist, for example, if a cable subscriber had pre-registered for a premium show. The cable system might recognize that registration, and authorize the user to view the show, by sending to the user an ID corresponding to the System ID.

When Rule 706 is invoked, corresponding Commands 707 access System ID 28 and obtain the system ID number. The commands then compare that number to Authorized System ID 708, specified by Rule 706. If the two numbers match, Commands 707 release Cryptographic Key 709 to Stream Controller 18, which uses Cryptographic Key 709 to decrypt the stream corresponding to ES_ID 2031. If the two numbers do not match, Commands 707 fail to release Cryptographic Key 709, so that Stream Controller 18 is unable to decrypt the stream.

In order to carry out these functions, in one embodiment, Control Block 13 includes, or has access to, a processing unit and memory. The processing unit is preferably capable of executing any of the commands which may be included or invoked by any of the rules. The memory will store the rules and association information (ID of the control message and IDs of any governed ESs).

Since the functions being carried out by Control Block 13 are sensitive, and involve governance of content which may be valuable, Control Block 13 may be partially or completely protected by a barrier which resists tampering and observation. As is described above, the processing unit, secure memory, and various other governance-related elements may be contained in IRP 22, which may be included in or separate from Control Block 13.

Control Block 13 may also carry out somewhat more complex operations. In one example, a control message may require that information from System 1 not only be accessed and compared to expected information, but stored for future use. For example, a control message might allow decryption of a Stream, but only after System ID 28 has been downloaded to and stored in Control Block 13. This would allow a control message to check the stored System ID against System ID 28 on a regular basis, or perhaps on every attempted re-viewing of a particular Stream, thereby allowing the control message to insure that the Stream is only played on a single System.

Control Block 13 may also obtain information dynamically. For example, System 1 may include User Interface 29, which can include any type of user input functionality (e.g., hardware buttons, information displayed on a video screen, etc.) A particular rule from a control message may require that the user enter information prior to allowing decryption or use of a stream. That information may, for example, be a password, which the Rule can then check against a stored password to insure that the particular user is authorized to render the stream.

Information obtained from the user might be more complicated. For example, a rule might require that the user input payment or personal information prior to allowing release of a cryptographic key. Payment information could, for example, constitute a credit card or debit card number. Personal information could include the user's name, age, address, email address, phone number, etc. Entered information could then be sent through Port 21 to External Server 30 for verification. Following receipt of a verification message from External Server 30, the Rule could then authorize release of a cryptographic key. Alternatively, Control Block 13 may be designed to operate in an "off-line" mode, storing the information pending later hookup to an external device (or network). In such a case, Control Block 13 might require that a connection be made at periodic intervals, or might limit the number of authorizations which may be obtained pending the establishment of an external connection.

In a somewhat more complex scenario, a control message may include conditional rules. One particular example is illustrated by row 4 of the table shown in FIG. 7, in which Control Message 700 is shown as controlling streams 49-53. Control Message 700 further consists of Rule 710, Commands 711 and Cryptographic Keys 712-716. There could, of course, be a number of additional cryptographic keys stored with the message.

In this case, Rule 710 specifies that a user who agrees to pay a certain amount (or provide a certain amount of information) may view Stream 49, but all other users are required to view Stream 50, or a combination of Streams 49 and 50. In this case, Stream 49 may represent a movie or television program, while Stream 50 represents advertisements. In one embodiment, different portions of Stream 49 may be decrypted with different keys so that, for example, a first portion is decrypted with Key 712, a second portion is decrypted with Key 713, a third portion is decrypted with Key 714, and so on. Rule 710 may include all keys used to decrypt the entirety of Stream 49. When the user initially attempts to access the video encoded in Stream 49, Rule 710 could put up a message asking if the user would prefer to use pay for view mode or advertising mode. If the user selects pay for view mode, Rule 710 could store (or transmit) the payment information, and pass Cryptographic Key 712 to Stream Controller 18. Stream Controller 18 could use Cryptographic Key 712 to decrypt the first stream until receipt of a header indicating that a different key is needed to decrypt the following set of packets. Upon request by Stream Controller 18, Control Block 13 would then check to determine that payment had been made, and then release Cryptographic Key 713, which would be used to decrypt the following packets, and so on. Rule 710 could additionally release Cryptographic Key 716, corresponding to Organization Stream 52, which corresponds to video without advertisements.

If, on the other hand, the user had chosen the advertising mode, Rule 710 could release Cryptographic Key 712 to Stream Controller 18 to allow decryption of Stream 49. Rule 710 could also authorize decryption of Stream 50 which contains the advertisements. Rule 710 could further release Cryptographic Key 715 to Organization Block 8. Cryptographic Key 715 matches Organization Stream 51. Organization Stream 51 references the video from Stream 49, but also references advertisements from Stream 50. Rule 710 would refuse to release Cryptographic Key 716, which corresponds to Organization Stream 52, which corresponds to the video without advertisements.

In operation, Control Block 13 could monitor information from Composite Block 11 over Control Line 16. That information could include the identity of each object actually rendered, as well as a start and stop time for the rendering. Control Block 13 could use this information to determine that an advertisement had actually been rendered, prior to releasing Cryptographic Key 713 for decryption of the second portion of video from Stream 49. This feedback loop allows Control Block 13 to be certain that the advertisements are not only being decrypted, but are also being displayed. This may be necessary because Composite Block 11 may be relatively unprotected, thereby allowing an unscrupulous user to remove advertisements before viewing.

A variety of additional relatively complex scenarios are possible. For example, rules from Control Block 13 could customize the programming for a particular geographic location or a particular type of viewer, by using information on the location or the viewer to control conditional decryption or use. This information could be stored in System 1 or entered by the user.

In another example, shown at row 5 of Array 717, Rule 719 may specify Budget 718, which may include information relating to the number of uses available to the user, the amount of money the user has to spend, etc. In operation, Rule 719 may require that

Budget 718 be securely stored and decremented each time a budgeted activity occurs (e.g., each time the associated work is played). Once the budget reaches zero, Rule 719 may specify that the work may no longer be played, or may display a message to the user indicating that the user may obtain additional budget by, for example, entering a credit card number or password, or contacting an external server.

In another example, a rule may control the ability of a user to copy a work to another device. The rule may, for example, specify that the user is authorized to use the governed work on more than one device, but with only one use being valid at any time. The rule may specify that an indication be securely stored regarding whether the user has "checked out" the work. If the user copies the work to another device (e.g., through Port 21), the rule may require that the work only be transmitted in encrypted form, and that the relevant control messages be transmitted along with it. The rule can further require that an indicator be securely set, and that the indicator be checked each time the user attempts to use or copy the work. If the indicator is set, the rule might require that the work not be decrypted or used, since the user only has the right to use the work on one device at a time, and the indicator establishes that the work is currently "checked out" to another device and has not been checked back in.

The receiving device may include the same type of indicator, and may allow the user to use the work only as long as the indicator is not set. If the user desires to use the work on the original device, the two devices may communicate, with the indicator being set in the second and reset in the first. This allows the work to be stored in two locations, but only used in one.

In another embodiment, the same result may be reached by copying the relevant control message from one device to the other, then erasing it from the original device. Because the control message includes keys used for decryption, this would insure that the work could only be used in one device at a time.

In one embodiment, this technique may be used to communicate digital media files (e.g., music, video, etc.) from a personal computer to a consumer electronics device without allowing the user to make multiple choices for simultaneous use. Thus, a larger, more sophisticated device (e.g., a personal computer), could download a file, then "check out" the file to a portable device lacking certain functions present in the personal computer (e.g., a hand-held music player).

Rules may also be used to specify that an initial user may transfer the file to another user, but only by giving up control over the file. Such rules could operate similarly to the technique described above for transferring a file from one device to another, or could require that the original file be entirely erased from the original device after the transfer.

Rules in Control Block 13 may be added or updated through at least two channels. New rules may be obtained through Control Stream 6. If a control message contains an identifier corresponding to a control message already present in Control Block 13, that control message (including contained rules) may overwrite the original control message. A new rule may, for example, be identical to an existing rule, but with a new time stamp and new keys, thereby allowing decryption of a stream which had been encrypted with multiple keys. System 1 may be designed so that certain rules may not be overwritable. This may be enforced by designating certain positions in Array 717 as non-overwritable, or by providing a flag or other indicator to show that a particular rule cannot be overwritten or altered. This would allow for certain types of superdistribution models, including allowing a downstream distributor to add rules without allowing the downstream distributor to remove or alter the rules added by upstream distributors.

In addition, new rules may be encoded into Organization Stream 3, Audio Stream 4, or Video Stream 5, in the form of a watermark or steganographic encoding.

New rules may also be obtained through Port 21. Port 21 may connect to an external device (e.g., a smart card, portable memory, etc.) or may connect to an external network (e.g., External Server 30). Rules may be obtained through Port 21 either in an ad hoc manner, or as a result of requests sent by Control Block 13. For example, Control Message 14 (FIG. 7, row 6) may include a rule specifying that a new rule be downloaded from a particular URL, and used to govern Stream 1201.

Control messages, including rules, may be encoded using secure transmission formats such as DigiBoxes. A DigiBox is a secure container means for delivering a set of business rules, content description information, content decryption information and/or content validation information. One or more DigiBoxes can be placed into the headers of the media content or into data streams within the media.

FIG. 12 illustrates one embodiment of the DigiBox format and the manner in which that format is incorporated into a control message. Control Message 1201 is made up of Control Message Header 1202 and Control Message Contents 1203. As is described elsewhere, Control Message Header 1202 may include information used by Demux 7 (FIG. 1) to appropriately route the message to Control Block 13.

Control Message Contents 1203 of Control Message 1201 consists of DigiBox 1204, and may also include additional information. DigiBox 1204 consists of DigiBox Header 1205, Rules 1206 and Data 1207. Rules 1206 may include one or more rules. Data

1207 may include various types of data, including ES_ID 1208, Cryptographic Key 1209, and Validation Data 1210. Data 1207 may also include cryptographic information such as a specification of the encryption algorithm, chaining modes used with the algorithm, keys and initialization vectors used by the decryption and chaining.

Initialization vectors contained within Data 1207 are similar to cryptographic keys, in that they constitute input to the original encryption process and therefore are necessary for decryption. In one well-known prior art embodiment, the initialization vectors may be generated by starting with a base initialization vector (a 64 bit random number) and xor'ing in the frame number or start time for the content item.

Validation Data 1210 contained within Data 1207 may include cryptographic has or authentication values, cryptographic keys for calculating keyed authentication values (e.g., message authentication codes), digital signatures, and/or public key certificates used in validating digital certificates.

Thus, the DigiBox may incorporate the information described above as part of the control message, including the rules, the stream ID and the cryptographic keys and values.

In an alternative embodiment, DigiBox Header 1205 may be designed so that it can be read by Demux 7 and routed to Control Block 13. In such an embodiment, DigiBox 1204 would itself constitute the entirety of the control message, thus obviating the need to nest DigiBox 1204 within Control Message 1201.

Some or all of the contents of DigiBox 1204 will generally be encrypted. This may include Rules 1206, Data 1207, and possibly some or all of Header 1205. System 1 may be designed so that a DigiBox may only be decrypted (opened) in a protected environment such as IRP 22. In an alternate embodiment, Control Block 13 may directly incorporate the functionality of IRP 22, so that the DigiBox may be opened in Control Block 13 without the necessity of routing the DigiBox to IRP 22 for processing. In one embodiment, the cryptographic key used to decrypt DigiBox 1204 may be stored in IRP 22 (or Control Block 13), so that the DigiBox can only be opened in that protected environment.

Rules 1206 are rules governing access to or use of DigiBox Data 1207. In one embodiment, these rules do not directly control the governed streams. Since Cryptographic Key 1209 can only be accessed and used through compliance with Rules 1206, however, Rules 1206 in fact indirectly control the governed streams, since those streams can only be decrypted through use of the key, which can only be obtained in compliance with the rules. In another embodiment, Data 1207 may include additional rules, which may be extracted from the DigiBox and stored in a table such as Array 717 of FIG. 7.

The rules governing access to or use of a DigiBox may accompany the DigiBox, (as shown in FIG. 12) or may be separately transmitted, in which event Rules 1206 would contain a pointer or reference to the rules used to access Data 1207. Upon receipt of a DigiBox, Control Block 13 may receive rules separately through Control Stream 6, or may request and receive rules through Port 21.

Pipelined Implementation

One potential drawback to the system illustrated in FIG.1 consists of the fact that the system introduces complexity and feedback into a pipelined system designed to render content in real time. The rendering pipeline generally consists of Demux 7, Organization Block 8 and AV Block 9, Composite Block 11 and Rendering Device 12. Because content is received in a streamed fashion, and must be rendered in real time, pipelined processing must occur in a highly efficient manner, under tight time constraints. A failure to process within the time available may mean that output to Rendering Device 12 may be interrupted, or that incoming Bit Stream 2 may overflow available buffers, thereby causing the loss of some portion of the incoming data.

An alternative embodiment of System 1 is designed to address these problems, although at a possible cost in the ability to use standard system components and a possible cost in overall system security. This alternative embodiment is illustrated in FIG. 11, which shows System 1101.

System 1101 is similar to System 1 from FIG. 1 in many respects. It receives Bit Stream 1102, which consists of Organization Stream 1103, Audio Stream 1104, Video Stream 1105 and Control Stream 1106. These streams are received by Demux 1107, which passes Organization Stream 1103 to Organization Block and passes Audio Stream 1104 and Video Stream 1105 to AV Block 1109. Organization Block 1108 and AV Block 1109 operate similarly to their counterparts in FIG. 1, and pass information to Composite Block 1110, which organizes the information into a coherent whole and passes it to Rendering Device 1111. Streams sent to Organization Block 1108 are decrypted and/or validated by Stream Flow Controller 1112, and streams sent to AV Block 1109 are decrypted and/or validated by Stream Flow Controller 1113.

System 1101 differs from System 1, however, in that control and feedback are distributed, and integrated directly into the processing and rendering pipeline. System 1101 thus lacks a separate control block, and also lacks a feedback path back from the Composite Block 1110.

In System 1101, control is exercised directly at Organization Block 1108 and AV Block 1109. As in System 1, cryptographic keys are received through Control Stream 1106

(in an alternative embodiment, the keys could be incorporated directly into header or other information in Organization Stream 1103 or Audio/Video Streams 1104 and 1105). Those keys are included in a data format which includes information regarding the stream type of the encrypted content and, if multiple stream types are possible, an identifier for the particular controlled stream.

When Demux 1107 encounters a key in Control Stream 1106, it reads the information relating to the stream type, and routes the key to the appropriate stream flow controller. If Demux 1107 encounters a key designated for decryption or validation of Organization Stream 1103, for example, it routes that key to Stream Flow Controller 1112.

Stream Flow Controller 1112 stores received keys in Storage Location 1114. Storage Location 1114 stores the keys and also stores an indicator of the controlled stream ID.

Stream Flow Controller 1112 includes Cryptographic Engine 1115, which uses the received keys to decrypt and/or validate encrypted and/or protected portions of Organization Stream 1103. The keys may themselves be received in an encrypted manner, in order to provide some degree of security. In such a case, Stream Flow Controller may use a variety of techniques to decrypt the key, including using stored information as a key, or as a key seed. That stored information could, for example, constitute a "meta-key" provided earlier through Bit Stream 1102 or through a separate port.

Stream Flow Controller 1113, associated with AV Block 1109, contains a corresponding Storage Location 1116 and Cryptographic Engine 1117, and operates in a manner similar to the operation described for Stream Flow Controller 1112.

This implementation avoids the latency penalty which may be inherent in the necessity for communication between stream flow controllers and a separate control block.

This alternate implementation may also eliminate the feedback channel from the composite block (FIG.1, Control Line 16). This feedback channel may be used in order to insure that the content being passed from Composite Block 11 to Rendering Device 12 is content that has been authorized for rendering. In the alternate embodiment shown in FIG.11, this feedback channel does not exist. Instead, this implementation relies on the fact that Composite Block 1110 depends upon information from Organization Block 1108 to determine the exact structure of the information being sent to Rendering Device 1111. Composite Block 1110 cannot composite information in a manner contrary to the organization dictated by Organization Block 1108.

In one embodiment, this control by Organization Block 1108 may be sufficient to obviate the need for any feedback, since Organization Block 1108 may be designed so that it accepts information only through Stream Controller 1112, and Stream Controller 1112 may be designed so that it only decrypts or validates information under the control of rules stored in Storage Location 1114.

In such an embodiment, security may be further increased by incorporating Secure Memory 1118 into Organization Block 1108. Secure Memory 1118 may store a copy or hash of the organization tree validly decrypted by Stream Controller 1112, and in current use in Main Organization Block Memory 1119. Organization Block 1108 may be used to periodically compare the organization tree stored in Main Organization Block Memory 1119 to the tree stored in Secure Memory 1118. If a discrepancy is spotted, this may indicate that an attacker has altered the organization tree stored in Main Organization Block 1119, thereby possibly allowing for the rendering of content in violation of applicable rules. Under such circumstances, Organization Block 1108 may be used to take protective measures, including replacing the contents of Main Organization Block Memory 1119 with the contents of Secure Memory 1118.

MPEG-4 Implementation

The generic system described above may be embodied in an MPEG-4 system, as illustrated in FIG. 8, which shows MPEG-4 System 801.

MPEG-4 System 801 accepts MPEG-4 Bit Stream 802 as input. MPEG-4 Bit Stream 802 includes BIFS Stream 803, OD Stream 804, Audio Stream 805, Video Stream 806 and IPMP Stream 807. These streams are passed to Demux 808, which examines header information and routes packets as appropriate, to BIFS 809, AVO 810, OD 811 or IPMP System 812.

IPMP System 812 receives IPMP messages through IPMP Stream 807. Those messages may include header information identifying the particular message, as well as an associated IPMP message. The IPMP message may include control information, which may include a cryptographic key, validation information, and/or may include complex governance rules, as are described above.

Stream Controllers 813, 814 and 815 act to decrypt, validate, and/or govern streams passed to BIFS 809, AVO 810 and OD 811, respectively.

OD 811 holds object descriptors, which contain metadata describing particular objects. This metadata includes an identifier of the particular Elementary Stream or streams which include the object, and may also include a pointer to a particular IPMP message which governs the object. Alternatively, the relationship between IPMP messages and particular objects or streams may be stored in a table or other form within IPMP System 812.

IPMP System 812 may exercise control over other functional blocks through Control Lines 816, 817, 818 and 819, each of which may transmit control/governance signals from IPMP System 812 and information or requests from other functional blocks to IPMP System 812. The information requests may include an ES_ID and a time stamp, which IPMP System 812 may use to determine which particular message (e.g., key) should be used and when.

In an alternative embodiment, IPMP System 812 may exercise control over Composite and Render 821 by receiving a hash of the currently valid BIFS tree (possibly through IPMP stream 807), and periodically checking the hash against the BIFS tree stored in BIFS 809. Because BIFS 809 controls the manner in which Composite and Render 821 renders information, if IPMP System 812 confirms that the current BIFS tree is the same as the authorized tree received through BIFS Stream 803, IPMP System 812 can confirm that the proper content is being rendered, even without receiving feedback directly from Composite and Render 821. This may be necessary, since BIFS 809 may communicate with Port 822, which may allow a user to insert information into BIFS 809, thereby creating a possibility that a user could insert an unauthorized BIFS tree and thereby gain unauthorized access to content.

When a stream controller receives encrypted or otherwise governed information, it may send the ES_ID and time stamp directly to IPMP System 812. Alternatively, it may send this information to OD 811, which may reply with the ID of the IPMP message which governs that object or stream. The stream controller can then use that IPMP message ID to request decryption, validation, and/or governance from IPMP System 812. Alternatively, OD 811 can pass the IPMP ID to IPMP System 812, which can initiate contact with the appropriate stream controller.

IPMP System 812 may obtain IPMP information through two channels other than IPMP Stream 807. The first of these channels is Port 820, which may be directly connected to a device or memory (e.g., a smart card, a DVD disk, etc.) or to an external network (e.g., the Internet). An IPMP message may contain a pointer to information obtainable through Port 812, such as a URL, address on a DVD disk, etc. That URL may contain specific controls needed by the IPMP message, or may contain ancillary required information, such as, for example, information relating to the budget of a particular user.

IPMP System 812 may also obtain IPMP information through OD updates contained in OD Stream 804. OD Stream 804 contains metadata identifying particular objects. A particular OD Message may take the format shown in FIG. 9. In this figure, OD Message 901 includes Header 902, which identifies the following packets as part of the OD stream, and indicates the number of packets. OD Message 901 further consists of Message 903, which includes a series of Pointers 904 and associated Metadata 905. Each Pointer 904 identifies a particular Elementary Stream, and the associated metadata is applicable to that stream. Finally, OD Message 901 may contain an IPMP Pointer 906, which identifies a particular IPMP message.

In aggregate, the information contained in OD Message 901 constitutes an object descriptor, since it identifies and describes each elementary stream which makes up the object, and identifies the IPMP message which governs the object. OD Message 901 may be stored in OD 811, along with other messages, each constituting an object descriptor.

Object descriptors stored in OD 811 may be updated through OD Stream 804, which may pass through a new object descriptor corresponding to the same object. The new object descriptor then overwrites the existing object descriptor. This mechanism may be used to change the IPMP message which controls a particular object, by using a new object descriptor which is identical to the existing object descriptor, with the exception of the IPMP pointer.

OD Stream 804 can also carry IPMP_DescriptorUpdate messages. Each such message may have the same format as IPMP messages carried on the IPMP stream, including an IPMP ID and an IPMP message.

IPMP_DescriptorUpdate messages may be stored in a table or array in OD 811, or may be passed to IPMP System 812, where they may overwrite existing stored IPMP messages, or may add to the stored messages.

Since IPMP information may be separately conveyed through the OD stream or the IPMP stream, MPEG-4 System 801 may be designed so that it only accepts information through one or the other of these channels.

In another embodiment, the existence of the two channels may be used to allow multi-stage distribution, with governance added at later stages, but with no risk that later alterations may override governance added at an earlier stage.

Such a system is illustrated in FIG. 10. In this Figure, IPMP System 812 includes IPMP Table 1002, which has slots for 256 IPMP messages. This table stores the IPMP_ID implicitly, as the location at which the information is stored, shown in Column 1003. The IPMP message associated with IPMP_ID 4, for example, is stored at slot 4 of IPMP Table 1002.

Each location in IPMP Table 1002 includes Valid Indicator 1004 and Source Indicator 1005. Valid Indicator 1004 is set for a particular location when an IPMP message is stored at that location. This allows IPMP System 812 to identify slots which are unfilled, which otherwise might be difficult, since at start-up the slots may be filled with random information. This also allows IPMP System 812 to identify messages which are no longer valid and which may be replaced. Valid Indicator 1004 may store time stamp information for the period during which the message is valid with IPMP System 812 determining validity by checking the stored time stamp information against the currently valid time.

Source Indicator 1005 is set based on whether the associated IPMP message was received from IPMP Stream 807 or from OD Stream 804.

These indicators allow IPMP System 812 to establish a hierarchy of messages, and to control the manner in which messages are added and updated. IPMP System 812 may be designed to evaluate the indicators for a particular location once a message is received corresponding to that location. If the valid indicator is set to invalid, IPMP System 812 may be designed to automatically write the IPMP message into that slot. If the valid indicator is set to valid, IPMP System 812 may then be designed to check the source indicator. If the source indicator indicates that the associated message was received through OD Stream 804, IPMP System 1812 may be designed to overwrite the existing message with the new message. If, however, the source indicator indicates that the associated message was received through IPMP Stream 807, IPMP System 812 may be designed to check the source of the new message. That check may be accomplished by examining the header associated with the new message, to determine if the new message was part of OD Stream 804 or part of IPMP Stream 807. Alternatively, IPMP System 812 may derive this information by determining whether the message was received directly from Demux 808 or through OD 811.

If the new message came through IPMP Stream 807, IPMP System 812 may be designed to store the new message in Table 1002, overwriting the existing message. If the new message came through OD Stream 804, on the other hand, IPMP System 812 may be designed to reject the new message.

This message hierarchy can be used to allow for a hierarchy of control. A studio, for example, may encode a movie in MPEG-4 format. The studio may store IPMP messages in the IPMP stream. Those messages may include a requirement that IPMP System 812 require that a trailer for another movie from the same studio be displayed prior to the display of the feature movie. IPMP System 812 could be used to monitor the beginning and end of rendering of the trailer (using feedback through Control Line 819) to ensure that the entire trailer plays, and that the user does not fast-forward through it.

The movie studio could encrypt the various elementary streams, including the IPMP stream. The movie studio could then provide the movie to a distributor, such as a cable channel. The movie studio could provide the distributor with a key enabling the distributor to decrypt the OD stream (or could leave the OD stream unencrypted), and the ability to insert new messages in that stream. The cable channel could, for example, include a rule in the OD stream specifying that the IPMP system check to determine if a user has paid for premium viewing, decrypt the movie if premium viewing has been paid for, but insert advertisements (and require that they be rendered) if premium viewing has not been paid for).

The cable channel would therefore have the ability to add its own rules into the MPEG-4 Bit Stream, but with no risk that the cable channel would eliminate or alter the rules used by the movie studio (e.g., by changing the trailer from a movie being promoted by the studio to a rival movie being promoted by the cable channel). The studio's rules could specify the types of new rules which would be allowed through the OD stream, thereby providing the studio a high degree of control.

This same mechanism could be used to allow superdistribution of content, possibly from one user to another. A user could be provided with a programming interface enabling the insertion of messages into the OD stream. A user might, for example, insert a message requiring that a payment of $1.00 be made to the user's account before the movie can be viewed. The user could then provide the movie to another user (or distribute it through a medium whereby copying is uncontrolled, such as the Internet), and still receive payment. Because the user's rules could not overrule the studio's rules, however, the studio could be certain that its rules would be observed. Those might include rules specifying the types of rules a user would be allowed to add (e.g., limiting the price for redistribution).

MPEG-4 System 801 may also be designed to include a particular type of IPMP system, which may be incompatible with IPMP systems that may be designed into other MPEG-4 systems. This may be possible because the MPEG-4 standard does not specify the format of the information contained in the IPMP stream, thereby allowing different content providers to encode information in differing manners.

IPMP System 812 in MPEG-4 System 801 may be designed for an environment in which differing IPMP formats exist. That system may scan the IPMP stream for headers that are compatible with IPMP System 812. All other headers (and associated packets) may be discarded. Such a mechanism would allow content providers to incorporate the same IPMP message in multiple formats, without any concern that encountering an unfamiliar format would cause an IPMP system to fail. In particular, IPMP headers can incorporate an IPMP System Type Identifier. Those identifiers could be assigned by a central authority, to avoid the possibility that two incompatible systems might choose the same identifier.

IPMP System 801 might be designed to be compatible with multiple formats. In such a case, IPMP System 801 might scan headers to locate the first header containing an IPMP System Identifier compatible with IPMP System 801. IPMP System 801 could then select only headers corresponding to that IPMP System Identifier, discarding all other headers, including headers incorporating alternate IPMP System Identifiers also recognized by the IPMP system.

Such a design would allow a content provider to provide multiple formats, and to order them from most to least preferred, by including the most preferred format first, the second most preferred format second, and so on. Since IPMP System 801 locks onto the first compatible format it finds, this ordering in IPMP Stream 801 would insure that the IPMP system chose the format most desired by the content provider.

Even if different IPMP formats are used, content will probably be encoded (and encrypted) using a single algorithm, since sending multiple versions of content would impose a significant bandwidth burden. Thus, ordinarily it will be necessary for content to be encrypted using a recognized and common encryption scheme. One such scheme could use the DES algorithm in output feedback mode.

This method of screening IPMP headers, and locking onto a particular format may also be used to customize an MPEG-4 bit Stream for the functional capabilities of a particular MPEG-4 system. Systems capable of rendering MPEG-4 content may span a considerable range of functionality, from high-end home theaters to handheld devices. Governance options suitable for one type of system may be irrelevant to other systems.

For example, MPEG-4 System 801 may include a connection to the Internet through Port 820, whereas a second MPEG-4 system (for example a handheld Walkman-like device) may lack such a connection. A content provider might want to provide an option to a viewer, allowing the viewer to see content for free in return for providing information about the viewer. The content provider could insert a rule asking the user whether the user wants to view the content at a cost, or enter identification information. The rule could then send the information through a port to the Internet, to a URL specified in the rule. A site at that URL could then evaluate the user information, and download advertisements targeted to the particular user.

Although this might be a valuable option for a content provider, it obviously makes no sense for a device which is not necessarily connected to the Internet. It would make no sense to present this option to the user of a non-connected device, since even if that user entered the information, the rule would have no way to provide the information to an external URL or download the advertisements. In such a case, the content provider might prefer to require that the user watch preselected ads contained in the original MPEG-4 bit stream.

Header information in the IPMP stream could be used to customize an MPEG-4 bit stream for particular devices. As with the IPMP System Type information, IPMP Header information could include MPEG-4 System Types. These could include 8 or 16-bit values, with particular features represented by bit maps. Thus, the presence of a bit at position 2, for example, could indicate that a device includes a persistent connection to the Internet.

An IPMP system could then evaluate the headers, and lock on to the first header describing functionality less than or equal to the functionality contained in the MPEG-4 device in which the IPMP system is embedded. If the header constituted a complete match for the functionality of the MPEG-4 device, the IPMP system could then cease looking. If the header constitutes less than a complete match (e.g., a header for a system which has an Internet connection, but lacks a digital output port, when the system includes both), the IPMP system can lock on to that header, but continue to scan for closer matches, locking on to a closer match if and when one is found.

The IPMP messages identified by a particular header would be those suited for the particular functionality of the MPEG-4 device, and would allow for customization of the MPEG-4 bit stream for that functionality. In the context of the example given above, the IPMP system for an MPEG-4 device containing an Internet connection would lock on to a particular header, and would download the IPMP messages characterized by that header. Those messages would prompt the user for information, would provide that information to the URL, and would authorize decryption and rendering of the movie, with the advertisements inserted at the appropriate spot.

In the case of an MPEG-4 device without an Internet connection, on the other hand, the IPMP system would lock onto a set of headers lacking the bit indicating an Internet connection, and would download the rules associated with that header. Those rules might not provide any option to the user. The rules might allow decryption of the content, but would also specify decryption of an additional ES from the MPEG-4 stream. That additional ES would contain the advertisements, and the IPMP system would require decryption and rendering of the advertisements, checking Control Line 819 to make certain that this had occurred. In the case of the system with the Internet connection, however, the rules allowing decryption and requiring rendering of the ES containing the advertisements would never be loaded, since those rules would be contained within messages identified by the wrong type of header. The advertisement ES would therefore never be decrypted and would be ignored by the MPEG-4 device.

FIG. 21 illustrates one manner in which a protected MPEG-4 file may be created. In this figure, CreateBox 2101 represents a DigiBox creation utility, which accepts keys and rules. In one embodiment, CreateBox 2101 may pass these keys and rules to IRP 2102 and receive DigiBox 2103 from IRP 2102. In another embodiment, IRP 2102 may be incorporated into CreateBox 2101, which accepts keys and rules and outputs DigiBox 2103.

DigiBox 2103 contains governance rules, initialization vectors and keys. DigiBox 2103 is passed from CreateBox 2101 to Bif Encoder 2104. Bif Encoder 2104 may be conventional, with the exception that it is designed to accept and process DigiBoxes such as DigiBox 2103. Bif Encoder 2104 also accepts a .txt file containing a scene graph, and initial object descriptor commands.

Bif Encoder 2104 outputs a .bif file, containing the scene graph stream (in compressed binary form) and a .od file, containing the initial object descriptor commands, the object descriptor stream, and DigiBox 2103.

Bif Encoder 2104 passes the .bif file and the .od file to Mux 2105. Mux 2105 also accepts compressed audio and video files, as well as a .scr file that contains the stream description. Mux 2105 creates IPMP streams, descriptors and messages, encrypts the content streams, interleaves the received streams, and outputs Protected MPEG-4 Content File 2106, consisting of Initial Object Descriptor 2107 and Encrypted Content 2108. Initial Object Descriptor 2107 contains DigiBox 2103, as well as other information. Encrypted Content 2108 may include a scene graph stream (i.e., a BIFS stream), an object descriptor stream, IPMP streams, and encrypted content streams.

If DigiBox 2103 contains all keys and rules necessary to render all of the content, it may be unnecessary for Mux 2105 to create any IPMP streams. If additional keys or rules may be necessary for at least a portion of the content, Mux 2105 may incorporate those rules and keys into one or more additional DigiBoxes, and incorporate those DigiBoxes either in the IPMP stream or in the OD update stream.

FIG. 22 illustrates one manner in which control may be incorporated into an existing MPEG-4 stream. In this figure, Unprotected MPEG-4 Content File 2201 includes Initial Object Descriptor 2202 and Content 2203. The content may include a scene description stream (or BIF stream), an object descriptor stream, a video stream, an audio stream, and possibly additional content streams.

Unprotected MPEG-4 Content File 2201 is passed to Repackager 2204, which also accepts keys and rules. Repackager 2204 passes the keys and rules to IRP 2205, and receives DigiBox 2206 in return, containing keys, rules and initialization vectors. In an alternate embodiment, IRP 2205 may be incorporated directly into Repackager 2204.

Repackager 2204 demuxes Unprotected MPEG-4 Content File 2201. It inserts DigiBox 2206 into the Initial Object Descriptor and encrypts the various content streams. Repackager 2204 also adds the IPMP stream, if this is necessary (including if additional DigiBoxes are necessary).

Repackager 2204 outputs Protected MPEG-4 Content File 2207, consisting of Initial Object Descriptor 2208 (including DigiBox 2206) and Encrypted Content 2209 (consisting of various streams, including the IPMP streams, if necessary).

Real Networks Implementation

In one embodiment, the elements described above may be used in connection with information encoded in compliance with formats established by Real Networks, Inc.

The Real Networks file format (RMFF) is illustrated in FIG. 13. This format includes a block of headers at the beginning (Header 1301), followed by a collection of content packets (Content 1302), followed by an index used for seek and goto operations (Index 1303). Each file can contain several streams of different types. For each stream, there is a "Media Properties Header" (1304) used to describe the format of the media content (e.g., compression format) and provide stream specific information (e.g., parameters for the decompressor).

Real Networks streams can be protected by inserting a DigiBox into Header 1301 and encrypting the data packets contained in Content 1302. The altered format is illustrated in FIG.14, which shows Header 1401, including Media Properties Headers 1402 and 1403, which in turn contain DigiBoxes 1404 and 1405, respectively. The format also includes encrypted Content 1406 and Index 1407.

In one embodiment, the declared type of the data is changed from the standard Real Networks format to a new type (e.g., RNWK_Protected.) The old type is then saved. Changing the type forces the Real Networks player to load a "Trust Plugin," since this Plugin is registered as the only decoder module that can process streams of type "RNWK-Protected." The Trust Plugin opens the DigiBox, gets approval from the user, if it is needed, determines the original content type, loads a decoder plugin for the original content, and then decrypts and/or validates the content, passing it to the content decoder plugin to be decompressed and presented to the user.

In one embodiment, the specific alterations made to the Real Networks file format are the following:

- Increase the preroll time to force larger buffers on playback. In a current embodiment, an increase of 3 seconds is used. Larger buffers are needed because of the extra steps needed to decrypt the content.
- Modify each stream-specific header by changing the mime type to "RNWK-Protected", saving the old mime type in the decoder specific information and adding a content identifier and DigiBox to the decoder specific information. The DigiBox contains the key, initialization vector (IV), version information, and watermarking instructions. The key, IV and content identifier are generated automatically, or can be provided as command-line parameters. The same key, IV and content identifier are used for every stream.
- Content packets are selectively encrypted. In one embodiment, content packets whose start time in milliseconds is in the first half-second of each 5 seconds (i.e., starttime % 5000 < 500) are encrypted. This encrypts approximately one-tenth of the content reducing encryption and decryption costs, and damages the content, sufficiently to prevent resale. The encryption algorithm can be DES using output-feedback mode or any similar algorithm. The initialization vector is computed for each packet by xoring the stream's IV with the packet's start time in milliseconds. Some information unique to the stream should also be xored into the IV. In one embodiment, the same IV is used for multiple packets whenever two or more streams have packets with the same start time. This usually happens for the first packet in each stream since they usually have start time 0. Other than the first packet, it is rare to have two packets have the same start time.

In one embodiment, these changes to the Real Networks file format are accomplished as is shown in FIG. 15. As is illustrated, RMFF file 1501 is formatted in the standard Real Networks RMFF format. This file is passed to Packager 1502. Also passed to Packager 1502 is Rights File 1503. Packager 1503 generates Protected RMFF File 1504, which includes various alterations as described above and as listed in FIG. 15, including the incorporation of one or more DigiBoxes in the header, encryption of the content, modification of the mime type, etc.

In one embodiment, the trust plugin described above is illustrated in FIGs. 16 and 17. FIG. 16 illustrates the standard Real Networks architecture. File 1601 (e.g., a streaming audio file in Real Networks format) is provided to Real Networks G2 Client

Core 1602. File 1601 may be provided to RealNetworks G2 Client Core 1602 from Server 1603, or through Direct Connection 1604.

Upon receipt of File 1601, Real Networks G2 Client Core 1602 accesses a rendering plugin appropriate to File 1601, based on information which is obtained from the header associated with File 1601. Rendering Plugins 1605 and 1606 are shown. If File 1601 is of a type which cannot be rendered by either Rendering Plugin 1605 or Rendering Plugin 1606, Real Networks G2 Client Core 1602 may attempt to access an appropriate plugin, e.g., by asking for the user's assistance or by accessing a site associated with the particular file type.

Rendering Plug-In 1605 or 1606 processes File 1601 in a conventional manner. This processing most likely includes decompression of File 1601, and may include other types of processing useful for rendering the content. Once this processing is complete (keeping in mind that the content is streamed, so that processing may be occurring on one set of packets at the same time that another set of packets is being rendered), File 1601 is passed back to Real Networks G2 Client Core 1602, which then passes the information to Rendering Device 1607. Rendering Device 1607 may, for example, be a set of stereo speakers, a television receiver, etc.

FIG. 17 illustrates the manner in which a trust plugin operates within the overall Real Networks architecture. Much of the architecture illustrated in FIG. 17 is the same as that illustrated in FIG. 16. Thus, File 1701 is provided to Real Networks G2 Client Core 1702 through Server 1703 or through Direct Connection 1704. The file is processed by Real Networks G2 Client Core 1702, using plugins, including Rendering Plugins 1705 and 1706, and is then passed to Rendering Device 1707.

FIG. 17 differs from FIG. 16 in its incorporation of Trust Plugins 1708 and 1709, and IRP 1710. When initially registered with Real Networks G2 Client Core 1702, Trust Plugins 1708 and 1709 inform Real Networks G2 Client Core 1702 that they can process content of type RNWK-Protected. Whenever Real Networks G2 Client Core 1702 encounters a stream of this type, it is then enabled to create an instance of the trust plugin to process the stream, e.g., Trust Plugin 1708. It then passes the stream to the trust plugin.

The stream passed to Trust Plugin 1708 may be in the format shown in FIG. 14. In such a case, Trust Plugin 1708 extracts DigiBox 1404 from Media Properties Header 1402. It also extracts the content id and original mime type from Media Properties Header 1402. The Trust Plugin first checks to see if any other stream with the same content identifier has been opened. If so, then DigiBox 1404 is not processed further. Instead, the key and IV from the box for this other stream are used. This avoids the time cost of opening a second box. Also, this ensures that a user is only asked to pay once even if there are multiple protected streams. By sharing content ids, keys, and IVs, several files can be played with the user only paying once. This is useful when SMIL is used to play several RMFF files as a single presentation.

In an alternate and possibly more secure embodiment, this check is not performed, and the key and IV from the current DigiBox are used even if another stream with the content identifier has already been opened.

If no other stream has been identified with the same content identifier, Trust Plugin 1708 passes DigiBox 1404 to IRP 1710. IRP 1710 may be a software process running on the same computer as Real Networks G2 Client Core and Trust Plugin 1708. IRP 1710 may run in a protected environment or may incorporate tamper resistance techniques designed to render IRP 1710 resistant to attack.

IRP 1708 may process DigiBox 1404 and extract a cryptographic key and an IV, which may then be passed to Trust Plugin 1708. Trust Plugin 1708 may then use this information to decrypt Encrypted Contents 1406.

Trust Plugin 1708 uses the original mime type information extracted from Media Properties Header 1402 to create an instance of the rendering plugin to be used for the content (e.g., Rendering Plugin 1705). Once this is done, Trust Plugin 1708 behaves like an ordinary rendering plugin to the Real Networks G2 Client Core 1702, in that Real Networks G2 Client Core 1702 passes streamed information to Trust Plugin 1708, which decrypts that information and passes it to Rendering Plugin 1705. From the perspective of Real Networks G2 Client Core 1702, Trust Plugin 1708 constitutes the appropriate rendering pluin, and the core is not aware that the information is being passed by Trust Plugin 1708 to a second plugin (e.g., Rendering Plugin 1705).

Similarly, from the point of view of Rendering Plugin 1705, Trust Plugin 1708 behaves like Real Networks G2 Client Core 1702. Thus although Rendering Plugin 1705 receives decrypted stream information from Trust Plugin 1708, Rendering Plugin 1705 operates exactly as if the information had been received directly from Real Networks G2 Client Core 1702. In this manner, content formatted for Rendering Plugin 1705 may instead be first processed by Trust Plugin 1708, without requiring any alteration to Real Networks G2 Client Core 1702 or Rendering Plugin 1705.

Trust Plugin 1708 may also perform other processing that may be helpful for security purposes. For example, Trust Plugin 1708 may watermark the decrypted file prior to passing it to Rendering Plugin 1705, keeping in mind that the watermark algorithm must be such that it will survive decompression of the file by Rendering Plugin 1705.

MP3 Embodiment

The techniques described above can also be applied to MP3 streaming content.

The MP-3 specification does not define a standard file format, but does define a bit stream, which is illustrated in FIG.18. In FIG. 18, MP-3 Bit Stream 1801 includes Content 1802. Content 1802 is divided into frames, shown as Frame 1803, Frame 1804 and Frame 1805. The dots between Frame 1804 and 1805 symbolize the fact that Content 1802 may include a large number of frames.

Each frame includes its own small header, shown in FIG. 18 as Headers 1806, 1807 and 1808.

Many MP3 players support a small trailer defined by the ID3 V1 specification, shown as Trailer 1809. This is a 128 byte trailer for carrying fields like artist, title and year, shown as Fields 1810, 1811 and 1812. The ID3 V1 trailer is ignored by players not designed to read such trailers, since it does not appear to be valid MP3 data.

FIG. 19 shows one embodiment of protection applied to the MP3 format. This protected format constitutes File 1908 and includes the following items:

- Unencrypted MP3 Content 1912. This is the first information encountered by a player, and will be rendered by any standard MP3 player. It can include a message to the user indicating that the content is protected and providing instructions as to how the content can be accessed (e.g., a URL for a trust plugin, instructions on payment mechanisms, etc.) Unencrypted MP3 Content 1912 may include a "teaser," consisting of an initial portion of the content (e.g., 30 seconds), which is rendered at no cost, thereby allowing a user to sample the content prior to making a decision to purchase it.

- Encrypted MP-3 Content 1901, which may include thousands of MP-3 frames. In one embodiment, the first eight frames out of every 32 frames are encrypted. Thus, one-quareter of the frames are rendered unuseable unless a player is able to decrypt them. In practice, this may render the content un-sellable or unuseable, without imposing excessive encryption or decryption costs. To further reduce encryption and decryption costs, only 32 bytes in each frame are encrypted. In a current embodiment, these are the first 32 bytes after the header and CRC information. In a different embodiment, a different 32 bytes may be encrypted in every frame. In a current embodiment, the content is encrypted with the DES using algorithm output-feedback mode. The initial IV for the file is randomly generated and then xored with the frame number to generate a unique IV for each frame.

Many alternate embodiments may exist, including encrypting more or less information, and using different encryption algorithms.

- ID3 V1 Trailer 1902, including 128 bytes.

- Content ID 1903, including 16 bytes. This is used by the player application to avoid opening DigiBoxes which it has already opened.
- DigiBox 1904, which may comprise approximately 18K bytes. It includes Key 1909, IV 1910 and Watermarking Instructions 1911. Watermarking Instructions 1911 may be used in a process of watermarking the associated content.
- Address 1905, which contains the address in the file of Content ID 1903 and consists of 4 bytes.
- Trust ID 1906, which identifies this trusted MP-3 file and consists of 16 bytes.
- ID3 V1 Trailer 1907, which is a copy of Trailer 1902.

A conventional MP3 player encountering File 1908 would be unable to render Content 1901, since at least a portion of that content is encrypted. Such a player would most likely read through to Trailer 1902 and cease processing at that point. A conventional player looking for the ID3 trailer information will seek to the end and find it.

FIG. 20 illustrates one embodiment of an MP3 player designed to process and render protected content. This figure shows MP3 Player 2001, which includes Buffer 2006 and Decompressor 2007, and renders content to Rendering Device 2008. In one embodiment, this is a modified version of a player distributed by Sonique.

Player 2001 obtains Protected MP3 File 2002 through any standard interface. Protected MP3 File 2002 may have the format illustrated in FIG. 19.

When Player 2001 is asked to play Protected MP3 File 2002, Player 2001 first calls Trust Plug-In 2003, which includes Approval Function 2009 and Decrypt Function 2005. Trust Plugin 2003 calls Approval Function 2009 to determine if Protected MP3 File 2002 is protected and whether authorization exists to play the file. Approval Function 2009 is first given a pointer to Protected MP3 File 2002. It then checks Protected MP3 File 2002 for the presence of Trust ID 1906. If Trust ID 1906 is not found, Approval Function 2009 returns an indicator that the file is not protected. Player 2001 then proceeds to render the file as a normal MP3 file.

If Trust ID 1906 is found, Approval Function 2009 checks Content ID 1903 to see if it matches the Content ID of a file that has already been opened.

If Protected MP3 File 2002 has not been previously opened, DigiBox 1904 is retrieved by Approval Function 2009, and is passed to IRP 2004, which may include software running in a protected environment, or incorporating tamper resistance. IRP 2004 attempts to open DigiBox 1904 in compliance with the rules associated with that DigiBox. One such rule may require, for example, that the user indicate assent to pay for use of the content. If DigiBox 1904 cannot be opened (e.g., the user refuses to pay) a value is returned to Approval Function 2009 indicating that the file is protected and may not be played.

If DigiBox 1904 is opened in compliance with applicable rules, the key and IV are retrieved and passed to Decrypt Function 2005. The key and IV are stored with the content id for later re-use and Decrypt Function 2005 is initialized. This may improve overall system performance, since it reduces the number of times a DigiBox must be opened. Each such action may introduce significant latency.

On the other hand, storing this information in unprotected memory may reduce overall system security. Security may be enhanced either by not storing this information (thereby requiring that each DigiBox be opened, even if the corresponding file has already been opened through another DigiBox), or by storing this information in a protected form or in a secure location.

The stored key, IV and content id are referenced when Approval Function 2009 first checks Content ID 1903 to determine if it matches the Content ID of an already opened file. If the new Content ID matches a stored Content ID, Decrypt Function 2005 is reinitialized using the stored key and IV corresponding to the matching content id and value indicating that this is a protected file for which play is authorized is returned to Approval Function 2009.

Once Protected MP3 File 2002 has been opened, each time Player 2001 needs a packet, Player 2001 reads it into Buffer 2006, strips off the header and CRC and passes the remaining data and a frame number to Decrypt Function 2005, which decrypts the frame if necessary, and returns it to Player 2001.

In a current embodiment, although audio content is encrypted, headers or trailers are not encrypted. This allows the Player 2001 to process information in headers or trailers without intervention from Approval Function 2009 or Decrypt Function 2005. This allows Player 2001 to place information such as playing time, artist and title into a playlist display, and initialize Decompressor 2007, without any action required from Trust Plugin 2003.

Commerce Appliance Embodiment

This section will describe an embodiment, comprising a Commerce Appliance architecture designed to allow persistent control of digital works in consumer electronics devices. Although this is described as a separate embodiment, it should be understood that the features of this embodiment may be combined with, or supplant, the features of any of the embodiments provided elsewhere in this description.

In one embodiment, this section will describe modifications to the MPEG-4 standard designed to support the association of persistent rules and controls with MPEG-4 content, as well as elements necessary for a Commerce Appliance to use such content. This is intended, however, merely as an example.

In one embodiment, shown in FIG. 23, each Commerce Appliance 2301 includes a CMPS ("Content Management and Protection System") 2302. Each CMPS is responsible for governing the use of controlled content, including decrypting the content and ensuring that the content is only used as permitted by associated rules.

Each governed digital work is associated with one or more CMPOs (Content Management Protection Object), e.g., CMPOs 2303. Each CMPO may specify rules governing the use of the digital work, and may include keys used to decrypt the work.

CMPOs may be organized in an hierarchical fashion. In one embodiment, a content aggregator (e.g., a cable channel, a web site, etc.) may specify a Channel CMPO ("CCMPO") used to associate certain global rules with all content present on that channel. Each independent work may in turn have an associated Master CMPO ("MCMPO") used to associate rules applicable to the work as a whole. Each object (or Elementary Stream, in MPEG-4) may have associated with it a CMPO containing rules governing the particular object.

In one exemplary application, Commerce Appliance 2301 may be an MPEG-4 player containing CMPS 2302. Upon receipt of a user command to play a particular work, CMPS 2302 may download a MCMPO associated with the work and obtain rules, which may include conditions required for decryption and viewing of the work. If the rules are satisfied, CMPS 2302 may use keys from the MCMPO to decrypt any Elementary Streams ("ES"), and may pass the decrypted ESs into the buffers. Composition and rendering of the MPEG-4 work may thereafter proceeds according to the MPEG-4 standard, except that any storage location or bus which may contain the work in the clear must be secure, and CMPS 2302 may have the ability to govern downstream processing, as well as to obtain information regarding which AVOs were actually released for viewing.

In a variation, the process of obtaining and governing the work may include downloading a CCMPO which applies rules governing this and other works. If rules contained in the CCMPO are satisfied, CMPS 2302 may obtain a key used to decrypt the MCMPO associated with the particular work to be viewed.

In another variation, a CMPO may be associated with each ES. In this variation, the MCMPO supplies one or more keys for decryption of each CMPO, and each CMPO may in turn supply a key for decryption of the associated ES.

Commerce Appliance 2301 is a content-rendering device which includes the capability of supporting distributed, peer management of content related rights by securely applying rules and controls to govern the use of content. Commerce Appliance 2301 may include general-purpose functions devoted to acquisition and managed rendering of content (e.g., a DVD (and/or any other optical disk format) player is able to play a DVD (and/or any other optical disk format) disk and output content to a television.) Commerce Appliance 2301 may make use of any of the means for protecting and using digital content on high capacity optical disk, in one non-limiting example, a DVD disk, as described in the aforementioned Shear patent application.

Commerce Appliance 2301 also includes special-purpose functions relating to other management and protection of content functions. These special-purpose functions may be supported by one or more embedded or otherwise included CMPS 2302 in the form of a single CMPS or a cooperative CMPS arrangement, and may include a user interface (e.g., User Interface 2304) designed to display control-related information to the user and/or to receive control-related information and directions from the user. Commerce Appliance 2301 may also be designed so that it is networkable with other Commerce Appliances (e.g., a set-top box connected to a DVD player and a digital television) and/or with other devices, such as a computer arrangement, which may also include one or more CMPSs.

An important form of Commerce Appliance specifically anticipates secure coupling on a periodic or continual fashion with a computer managed docking environment (e.g., a standalone computer or other computer managed device which itself may be a Commerce Appliance) where the one or more CMPSs of the Commerce Appliance interoperate with the docking environment to form a single user arrangement whose performance of certain functions and/or certain content usage events is enabled by such inter-operation through, at least in part, cooperation between CMPSs and content usage management information of the Commerce Appliance and the trust environment capabilities of the docking environment, (e.g., further one or more CMPSs and content usage management information, such as, for example, information provided by use of CI).

An exemplary Commerce Appliance may be designed to comply with the emerging MPEG-4 standard for the formatting, multiplexing, transmission, compositing, and rendering of video and other types of information.

Commerce Appliance 2301 may be any computing device, one non-limiting example of which is a Personal Computer (PC) that includes MPEG-4 software (and/or hardware) for rendering content. In accordance with the present invention, the PC may also use one or more CMPSs as described herein.

The commerce appliance function is not restricted to streamed channel content but may include various browser-type applications consisting of aggregated composite content such as still imagery, text, synthetic and natural video and audio and functional content such as applets, animation models and so on. these devices include browsers, set-top boxes, etc.

Content Management and Protection System (CMPS)

Each commerce appliance includes one or more CMPS (e.g., CMPS 2302). The CMPS is responsible for invocation and application of rules and controls, including the use of rules and controls to govern the manner in which controlled content is used.

Particular functions of CMPS 2302 include the following:

(a) Identification and interpretation of rules.

CMPS 2302 must determine which rules are to be applied, and must determine how those rules are to be interpreted in light of existing state information. In one embodiment, this requires that CMPS 2302 obtain and decrypt one or more CMPOs 2303 associated with a work.

(b) Identification of content associated with particular rules.

CMPS 2302 must determine which content is governed by particular one or more rules. This may be accomplished by obtaining information from one or more CMPOs 2303 and/or other CI. In one embodiment, a CCMPO may identify a set of works, a MCMPO may identify a particular work and a CMPO may identify a particular ES or Audio Visual Object ("AVO").

(c) Decryption of content as allowed by the rules.

CMPS 2302 may be designed so that all content is routed through CMPS 2302 for decryption, prior to reinsertion into the data flow required by the relevant standard. In the case of MPEG-4, for example, the output from Demux 2305 may be fed into CMPS 2302. CMPS 2302 may then decrypt the content and, if relevant rules and controls are satisfied, feed the content into the MPEG-4 buffers. From that point, the data flow associated with the content may be as described by MPEG-4.

(d) Control of content based on rules.

CMPS 2302 may be used to control usage of content after the initial decryption, for example, through the use of secure event management as described in the incorporated Ginter '333 patent application. In the case of MPEG-4 systems, this may require that CMPS 2302 exercise control over hardware and/or software which performs the following functions: demuxing (performed by Demux 2305), decompression/buffering/decode into AVOs (performed by Scene Descriptor Graph 2306, AVO Decode 2307 and Object Descriptors 2308), scene rendering (performed in Composite and Render 2309).

CMPS 2302 may also be used to control use and consequences according to: (1) generational copy protection rules such as the CGMS and/or SGMS standards; (2) various Conditional Access control methods, such as those proposed and/or implemented by NDS as described in MPEG-4 document M2959, DAVIC "Copyright Control Framework" document, and in other publications; (3) a Rights Management Language, such as those proposed in the Ginter '333 patent application and/or as described by U.S. Patent No. 5,638, 443 to Stefik, et al.; (4) use policies described in accordance with AT&T's Policy Maker, as described by Blaze, Feigenbaum, and Lacy; (5) the CCI layer bits for IEEE 1394 serial bus transmission as specified by the DTDG subgroup of the DVD Copy Protection Technical Working Group and/or as implemented by the Hitachi, Intel, Matsushita, Sony and Toshiba proposed standard (hereafter "the five company proposal"); (6) controls transmitted using any secure container technology such as, for example, IBM Cryptolope; (7) any other means for specifying use rules and consequences.

(e) Monitoring use of content.

CMPS 2302 may be used to monitor content to: (i) ensure that rules are being complied with; (ii) ensure that no attempts are being made to tamper with the system or protected content; and (iii) record information used by rules, including usage information needed for payment purposes.

(f) Updating user budgets.

CMPS 2302 may be used to update user or other budgets to reflect usage.

(g) Exhaust information.

CMPS 2302 may be used to output payment and usage information ("exhaust information") to external processes, including one or more Commerce Utility Systems.

(h) Hardware identification and configuration.

(i) Obtaining new, additional, and/or augmented rules from an external process, one non-limiting example of which is a Rights and Permission Clearinghouse as described in the incorporated Shear patent application.

(j) Receiving keys, digital credentials, such as certificates, and/or administrative information, from certifying authorities, deployment managers, clearinghouses, and/or other trusted infrastructure services.

(k) Securely sending and/or receiving user and/or appliance profiling and/or attribute information.

(l) Securely identifying a user or a member of a class of users who requests content and/or CMPO and/or CMPS usage.

(m) Securely certifying or otherwise guaranteeing the authenticity of application code, for example certifying within CMPO 2301 and/or CMPS 2302 that application code containing rules and/or other application information, such as information written in Java code for conditional execution within a Commerce Appliance, and/or that executes at least in part outside of CMPO 2301 and/or CMPS 2302, has not been altered and/or has been delivered by a guaranteed (e.g., trusted) party.

(n) Securely processing independently delivered CI, such as described in the incorporated Ginter '333 patent application, to perform content usage control that protects the rights of plural, independent parties in a commerce value chain.

(o) Securely performing watermarking (including, for example fingerprinting) functions, for example as described in the Ginter '333 patent application and as incorporated herein, for example including interpreting watermarking information to control content usage and/or to issue an event message, wherein such event message may be reported back to a remote authority, such as, for example, a MCMPO rights clearinghouse management location.

CMPS 2302 may be used to identify and record the current hardware configuration of the Commerce Appliance and any connected devices (e.g., which loudspeakers are available, identification of attached monitors, including whether particular monitors have digital output ports, etc.) If attached devices (such as loudspeakers) also include CMPSs, the CMPSs may be used to communicate for purposes of coordination (e.g., a CMPS in a set-top box and/or loudspeaker arrangement may communicate with a CMPS in a downstream digital television or other display device to establish which CMPS will be responsible for governance or the nature of cooperative governance through a virtual rights process, said process optionally involving a rights authority server that may find, locate, provide, aggregate, distribute, and/or manage rights processes, such as described in the aforementioned Shear patent application, for employing plural CMPSs, for example, for a single user content processing and usage arrangement).

The present invention includes arrangements comprising plural Commerce Appliances and/or CMPSs in one or more user locations, non-limiting examples of which include a home, apartment, loft, office, and/or vehicle, such as a car, truck, sports utility vehicle, boat, ship, or airplane, that may communicate among themselves at least occasionally and may comprise a virtual network that operates in a logically cooperative manner, through at least in part the use of such CMPSs, to ensure optimal commercial flexibility and efficiency and the enforcement of rights of commerce value chain participants, including financial and copyright rights of providers, infrastructure rights of appliance providers, societal rights of government and/or societal bodies, and privacy rights of all parties, including consumers. Information related to interaction among such a network of value chain participants, including content usage auditing, content usage consequence, and CI specification, can be securely, variably reported to parties having right to such information, through, at least in part, use of such CMPSs, for example, as described in the aforementioned Ginter '712 patent application regarding the information reporting functioning of VDE nodes.

In one embodiment, shown in FIG. 24, CMPS 2401 consists of special-purpose hardware and resident software or firmware. These include the following:

(a) One or more processors or microcontrollers e.g. CPU 2402. CPU 2402 controls the overall processing of CMPS 2401, including execution of any necessary software.

(b) One or more external communications ports, e.g., Port 2403. Port 2403 communicates with External Network 2404, which may include LANs, WANs or distributed networks such as the Internet. External communications ports may also include one or more IEEE 1394 serial bus interfaces.

(c) Memory 2405. Types of memories which may be included in Memory 2405-- and examples of the information they may store -- are the following:

i. ROM 2406. ROM 2406 may include any information which is permanently stored in CMPS 2401, such as (1) CMPS Operating System 2407 and/or CMPS BIOS 2408, (2) Rules/Controls 2409 which are permanently stored in the CMPS; (3) Control Primitives 2410 which may be used to build rules or controls; (4) Keys 2411 associated with the CMPS, including a Public/Private Key Pair; (5) one or more Certificates 2412 designed to identify CMPS 2401 and/or the device, including version information; (6) Hardware Signature Information 2413 used to check for tampering (e.g., a hashed signature reflecting the expected hardware state of the device).

ii. RAM 2414. RAM 2414 may hold current state information needed by CMPS 2401, as well as information temporarily stored by CMPS 2401 for later use. Information stored in RAM 2414 may include the following: (1) Software 2415 currently executing in CPU 2402; (2) CMPOs 2416 which are currently active; (3) Content Object Identification 2417 of those content objects which are currently active (in an MPEG 4 system this would constitute, for example, an identification of active AVOs); (4) Rules 2418 which are currently active; (5) State Information 2419 regarding the current state of use of content, including an identification of any higher-order organization (in an MPEG-4 system this would constitute an identification of the scene descriptor tree and the current state of composition and rendering); (6) Stored Exhaust Information 2420 relating to use and/or the user, designed for external transmission; (7) Updated Budget Information 2421; (8) Content 2422; (9) Active Content Class Information 2423; and (10) Active User Identification 2424, including identification characteristic information.

iii. NVRAM 2425 (e.g., flash memory). This type of memory may hold information which is persistent but changeable, including at least some: (1) Budget Information 2426; (2) User Information 2427, such as identification, credit card numbers; preferred clearinghouses and other Commerce Utility Systems; (3) User Preferences 2428, such as preferences, profiles, and/or attribute information; and (4) Appliance Information 2429, such as attribution and/or state information.

The types of information described above and stored in CMPS Memory 2405 may be stored in alternative of the above memory types, for example, certain budget information may be located in ROM, information regarding specific one or more clearinghouses may be stored in ROM, certain active information may be moved into NVRAM, etc.

Budget information may include stored budgets made up of, for example:

(1) electronic cash;

(2) pre-authorized uses (e.g., based on a prepayment, the user has the right to watch 12 hours of programming).

(3) Security budgets related to patterns reflecting abnormal and/or unauthorized usage, for example, as described in the incorporated Shear patent, wherein such budgets restrict and/or report certain cumulative usage conduct.

(4) electronic credit, including credit resulting from usage events such as attention to promotional material and/or the playing of multiple works from one or more classes of works (e.g., certain publisher's works) triggering a credit or cash refund event and/or a discount on future playing of one or more of such publisher's works, such as other works provided by such publisher.

User information may include the following types of information for one or more authorized users of the Commerce Appliance:

(1) Name, address, telephone number, social security number or other identifier

(2) Information used to authenticate the user, which may include a user selected password and/or biometric data, such as fingerprints, retinal data, etc.

(3) User public/private key pair

(4) User attribute and/or profiling information.

iv. Removable Memory 2430. This may include any type of removable memory storage device, such as smart cards, floppy disks or DVD disks. If the commerce appliance is designed to play content received on removable memory devices (e.g., a DVD player), that capability may be used for purposes of the CMPS.

Memory 2405 may include a protected database, in which certain control, budget, audit, security, and/or cryptographic information is stored in secure memory, with complete information stored in an encrypted fashion in unsecure memory.

(d) Encryption/Decryption Engine 2431. CMPS 2401 must include a facility for decrypting received information, including content and CMPOs and/or other. CMPS 2401 may also include a facility for encrypting information if such information is to be transmitted outside the secure boundaries of CMPS 2401. This may include exhaust sent to clearinghouses or other external repositories; and content sent across unsecured buses for usage, such as content sent across IEEE 1394 Serial Bus 2432 to a computer central processing arrangement or to a viewing device such as a monitor, wherein a receiving CMPS may be employed to control such content's usage, including, for example, decrypting such content, as appropriate. Encryption/Decryption Engine 2431 may include a Random Number Generator 2433 used for the creation of keys or key pairs that can be used to identify and assure the uniqueness of CMPSs and support the opening of secure communication channels between such secure content control secure encryption/decryption arrangements.

(e) Secure Clock/Calendar 2434. CMPS 2401 may include Secure Clock/Calendar 2434 designed to provide absolute information regarding the date and time of day, information regarding elapsed absolute time, and/or relative timing information used to determine the elapsed time of operations performed by the system. Secure Clock/Calendar 2434 may include Battery Back Up 2435. It may further include Sync Mechanism 2436 for synchronization with outside timing information, used to recover the correct time in the event of a power loss, and/or to check for tampering.

(f) Interface 2437 to blocks used for content rendering and display. This interface is used for controlling rendering and display, based on rules, and for obtaining feedback information, which may be used for budgeting purposes or for providing information to outside servers (e.g., information on which content was actually displayed, which choices the user invoked, etc.) In the case of an MPEG-4 player such as is shown in

FIG. 23, this may include control over Commerce Appliance circuitry which handles, for example, buffering, the scene descriptor graph, AVO decode, object descriptors and composite and rendering (e.g., Control Lines 2310, 2311 and 2312).

Feedback Path 2313 from Composite and Render block 2309 may allow CMPS 2302 to determine whether and when content has actually been released to the viewer. For example, Composite and Render block 2309 can issue a start event to CMPS 2302 when an AVO object is released for viewing, and can issue a stop event to CMPS 2302 when the AVO object is no longer being viewed.

Feedback from Composite and Render block 2309 may also be used to detect tampering, by allowing CMPS 2302 to match the identification of the objects actually released for viewing with the identification of the objects authorized for release. Start and end time may also be compared with the expected elapsed time, with a mismatch possibly indicative of the occurrence of an unauthorized event.

In one embodiment, the following protocol may be used for feedback data:

start <id>, T, <instance number><clock time><rendering options>

Sent if elementary stream <id> is reachable in the SD-graph at time *T*, but not at time *T-1*.

end <id>, T, <instance number><clock time><rendering options>

T constitutes presentation time, clock time constitutes the wall clock time, including day and date information, and rendering options may include such information as QoS and rate of play (e.g., fast forward).

Sent if elementary stream **<id>** is reachable in the SD-graph at time *T-1* but not at time *T*. A SD-graph stream is reachable if, during traversal of the SD-graph for display update, the renderer encounters a node that the SD-graph update stream <id> created or modified. This implies that all nodes in the tree need an update history list. This list need not be as large as the number of streams. Further, it can be labeled to indicate if the CMPS will be watching for stream, if not labeled it will not record them. An AV elementary stream is reachable if the stream's content was rendered.

For SD-graph update streams, the object instance number is ignored. For AV streams, the instance number can be used to disambiguate the case where the display shows two or more instances of the same data stream simultaneously. Instance numbers do not have to count up. In this case, they are simply a unique id that allows the CMPS to match a start event with an end event.

In a second embodiment, CMPS 2302 may include some special purpose hardware in combination with general purpose hardware which is also used for other functions of the device. In this embodiment, care must be taken to ensure that commercially trusted CMPS functions are performed in a secure and tamper-resistant manner, despite the use of general purpose hardware. Each of the elements recited above may include dedicated CMPS functions and general purpose device functions:

(a) CPU/microcontroller. This may include one or more devices. If more than one device is included (e.g., a CPU and a DSP, a math coprocessor or a commerce coprocessor), these devices may be included within the same package, which may be rendered tamper-resistant, or the devices may communicate on a secure bus. The CPU may include two modes: a secure CMPS mode, and an unsecure general purpose mode. The secure CMPS mode may allow addressing of secure memory locations unavailable to the processor in general purpose mode. This may be accomplished, for example, by circuitry which remaps some of the available memory space, so that, in unsecure mode, the CPU cannot address secure memory locations.

(b) External communications ports. If the device, for example, a Commerce Appliance, is capable of receiving content or other information through a communications port (e.g., a cable connection, an Internet connection), this communications port can be used for CMPS purposes. In such a case, CMPS accesses to the external communications port is preferably designed to avoid or minimize interference with the use of such port for receipt of content.

(c) Memory. In some applications and embodiments, it is possible to operate a Commerce Appliance without NVRAM, wherein information that may be needed for CMPS operation that would employ NVRAM would be loaded into RAM, as required. ROM, RAM and NVRAM may be shared between CMPS uses and general uses. This can be accomplished in any of the following ways, or in a combination of these ways: (1) Some memory space may be rendered off-limits to general purpose uses, for example by remapping; (2) the entirety of the memory may be rendered secure, so that even portions of the memory being used for non-secure purposes cannot be observed or changed except in a secure and authorized manner; (3) CMPS information may be stored in an encrypted fashion, though this requires at least some RAM to be secure, since the CMPS will require direct access to unencrypted information stored in RAM.

(d) Encryption/decryption engine. Encryption and decryption functions, including key generation, may be handled by special purpose software running on a general purpose processor arrangement, particularly, for example, a floating point processor or DSP arrangement. That processor arrangement may also be used for purposes of decompressing and displaying content and/or for handling watermarking/fingerprinting insertion and/or reading. Alternatively, the device may include native encryption and decryption functions. For example, various emerging standards may require at least some degree of encryption and decryption of content designed to be passed across unsecure buses within and among devices such as DVD players, such as the "five company proposal" and other IEEE 1394 related initiatives. Circuitry designed to perform such encryption and decryption may also be usable for CMPS applications.

(e) Secure clock/calendar. The underlying device may already require at least some clock information. MPEG-4, for example, requires the use of clock information for synchronization of Elementary Streams. A secure CMPS clock can also be used for such purposes.

In a third embodiment, CMPS 2302 can be primarily software designed to run on a general purpose device which may include certain minimal security-related features. In such a case, CMPS 2302 may be received in the same channel as the content, or in a side-band channel. An I-CMPO and/or other CI may specify a particular type of CMPS, which Commerce Appliance 2301 must either have or acquire (e.g., download from a location specified by the I-CMPO), or CMPS 2302 may be included, for example, with an I-CMPO.

A software CMPS runs on the CPU of the Commerce Appliance. This approach may be inherently less secure than the use of dedicated hardware. If the Commerce Appliance includes secure hardware, the software CMPS may constitute a downloadable OS and/or BIOS which customizes the hardware for a particular type of commerce application.

In one embodiment, a software CMPS may make use of one or more software tamper resistance means that can materially "harden" software. These means include software obfuscation techniques that use algorithmic means to make it very difficult to reverse engineer some or all of a CMPS, and further make it difficult to generalize from a reverse engineering of a given one or more CMPS. Such obfuscation is preferably independent of source code and object code can be different for different CMPSs and different platforms, adding further complexity and separation of roles. Such obfuscation can be employed "independently" to both CI, such as an CMPO, as well as to some or all of the CMPS itself, thus obscuring both the processing environment and executable code for a process. The approach is also applicable for integrated software and hardware implementation CMPS implementations described above. Other tamper resistance means can also be employed, including using "hiding places" for storing certain state information in obscure and unexpected locations, such as locations in NV memory used for other purposes, and data hiding techniques such as watermarking/fingerprinting.

Association of CMPS With a Commerce Appliance

A CMPS may be permanently attached to a particular device, or may be partially or fully removable. A removable CMPS may include software which is securely loaded into a Commerce Appliance, and/or removable hardware. A removable CMPS may be personalized to one or more particular users, including user keys, budget information, preferences, etc., thereby allowing different users to use the same Commerce Appliance without commingling budgets and/or other rights, etc.

A CMPS may be designed for operation with certain types of content and/or for operation with certain types of business models. A Commerce Appliance may include more than one type of CMPS. For example, a Commerce Appliance designed to accept and display content pursuant to different standards may include one CMPS for each type of format. In addition, a Commerce Appliance may include a CMPS provided by a particular provider, designed to preferentially display certain types of content and to preferentially bill for such content through a particular channel (e.g., billing to one or more particular credit cards and/or using a particular one or more clearinghouses).

Source of Rules

The CMPS must recognize those rules which are to be applied to particular content. Such rules may be received by the CMPS from a variety of sources, depending on the particular embodiment used:

(a) CMPO. The rules may be included within a CMPO (e.g., CMPO 2303) and/or other CI. The CMPO and/or other CI may be incorporated within a content object or stream (as, e.g., a header on an MPEG-4 ES), and/or may be contained within a dedicated content object or stream encoded and received as per the underlying standard (e.g., an MPEG-4 CMPO ES), and/or may be received outside the normal content stream, in which event it may not be encoded as per the underlying standard (e.g., a CMPS received as an encrypted object through a sideband channel).

(b) CMPS. Rules may be permanently and/or persistently stored within a CMPS, e.g., Rules 2409. A CMPS may include default rules designed to handle certain situations, for example, where no CMPO and/or other necessary CI is received (e.g., content encoded under an earlier version of the standard which did not incorporate CMPOs, including MPEG-4 version 1). Complete rules which are stored within the CMPS may be directly or indirectly invoked by a CMPO and/or other CI. This may occur through the CI identifying particular rules through a pointer, and/or it may occur through the CI identifying itself and the general class of control it requires, with the CMPS then applying particular rules specific to that CMPS.

Rule "primitives" may also be stored within the CMPS (e.g., Control Primitives 2410). The CMPO and/or other CI may invoke these primitives by including a sequence of macro-type commands, each of which triggers a sequence of CMPS primitives.

(c) User. The user may be given the ability to create rules relating to the particular user's preferences. Such rules will generally be allowed to further restrict the use of content, but not to expand the use of content beyond that which would otherwise be allowed. Examples include: (a) rules designed to require that certain types of content (e.g., adult movies) only be accessible after entry of a password and/or only to certain CMPS users (e.g. adults, not children, as, for example, specified by parents and/or a societal body such as a government agency); (b) rules designed to require that only particular users be allowed to invoke operations requiring payment beyond a certain limit and/or aggregate payment over a certain amount.

The user may be allowed to create templates of rules such as described in the aforementioned Ginter '333 patent application (and incorporated herein). In addition, a CMPS arrangement, and/or a particular CMPO and/or other CI, may restrict the rules the user is allowed to specify. For example, a CI may specify that a user can copy a work, but cannot add rules to the work restricting the ability of a recipient to make additional copies (or to be able to view, but only after a payment to the first user). User supplied one or more rules may govern the use of – including privacy restrictions related to -- payment, audit, profiling, preference, and/or any other kind of information (e.g., information result as a consequence of the use of a CMPS arrangement, including, for example, use of secured content). Such user supplied one or more rules can be associated with the user and/or one or more Commerce Appliances in a user arrangement, whether or not the information is aggregated according to one or more criteria, and whether or not user and/or appliance identification information is removed during aggregation and/or subsequent reporting, distribution, or any other kind of use.

The ability to allow the user to specify rules allows the CMPS to subsume (and thereby replace) V-chips, since a parent can use content rating information to specify precisely what types of information each viewer will be allowed to watch (e.g., violent content can only be displayed after entry of a certain password and/or other identifier, including, for example, insertion of a removable hardware card (smart or rights card) possessed by a user).

(d) External network source. The rules may be stored on an external server. Rules may be addressed and downloaded by the CMPS if necessary (e.g., either the CMPO and/or other CI and/or the CMPS contains a pointer to certain rules location(s), such as one or more URLs). In addition, content providers and/or clearinghouses may broadcast rules designed for general applicability. For example, a content provider might broadcast a set of rules providing a discount to any user participating in a promotional event (e.g., by providing certain user information). Such rules could be received by all connected devices, could be received by certain devices identified as of interest by the content provider (e.g., all recent viewers of a particular program, as identified by exhaust information provided by the CMPS to a clearinghouse and/or all members having certain identity characteristics such as being members of one or more classes) and/or could be posted in central locations.

Example Embodiment

In one embodiment, a set of MPEG-4 Elementary Streams may make up a work. The Elementary Streams may be encrypted and multiplexed together to form an Aggregate Stream. One or more CMPOs may be present in such stream, or may otherwise be associated with the stream. Options are as follows:

1. Content may be streamed or may be received as static data structures.

2. A Work may be made up of a single stream or data structure, or of many separately addressable streams or data structures, each of which may constitute an Object.

3. If a Work is made up of separately addressable streams or data structures, those streams or data structures may be multiplexed together into an Aggregate Stream, or may be received separately.

4. If streams or data structures are multiplexed together into an Aggregate Stream, the streams or data structures may be encrypted prior to such multiplexing. The Aggregate Stream itself may be encrypted, whether or not the underlying streams or data structures are encrypted. The following possibilities therefore exist: (a) individual streams/data structures are unencrypted (in the clear), the Aggregate Stream is unencrypted; (b) individual streams/data structures are unencrypted prior to multiplexing, the Aggregate Stream is encrypted following multiplexing; (c) individual streams/data structures are encrypted prior to multiplexing, the Aggregate Stream is not encrypted following multiplexing; or (d) individual streams/data structures are encrypted prior to multiplexing, the Aggregate Stream is encrypted following multiplexing.

5. A CMPO may be associated with a channel (CCMPO), a work (MCMPO) or an individual Object (CMPO).

6. A CMPO may be received prior to the controlled data, may be received contemporaneously with the data, or may be received after the data (in which event use of the data must wait until the CMPO has been received).

7. A CMPO may be received as part of an Aggregate Stream or separately.

8. If a CMPO is received as part of the Aggregate Stream, it may be multiplexed together with the individual streams or data structures, or may constitute a separate stream or data structure.

9. If a CMPO is multiplexed within the Aggregate Stream, it may be encrypted or nonencrypted. If encrypted, it may be encrypted prior to multiplexing, and/or encrypted after multiplexing, if the entire Aggregate Stream is encrypted.

10. If a CMPO is received as part of the Aggregate Stream, it may be (a) a part of the stream or data structure which holds the content (e.g., a header); (b) a separate stream or data structure encoded pursuant to the same format as the streams or data structures which hold the content (e.g., an MPEG-4 ES) or (c) a separate stream or data structure encoded under a different format designed for CMPOs.

11. If a CMPO is a part of the stream or data structure which holds the content, it may be (a) a header which is received once and then persistently maintained for control of the content; (b) a header which is received at regular intervals within the stream or data structure; or (c) data distributed throughout the stream or data structure.

These various scenarios give rise to different requirements for demultiplexing and decryption of the CMPOs. FIG. 25 illustrates the following embodiment:

1. Aggregate Stream 2501 is made up of multiplexed ESs (e.g., ES 2502 and 2503). A combination of such ESs makes up a single work. Aggregate Stream 2501 is generated by a cable aggregator and received by a user's set-top box as one of a number of channels.

2. CCMPOs 2504 corresponding to each channel are sent along the cable in Header 2505 at regular intervals (e.g., once per second). When the set-top box is turned on, it polls each channel, and downloads all current CCMPOs. These are stored persistently, and are changed only if a new CCMPO is received which differs from prior CCMPOs.

3. When the user selects a channel, the set-top box addresses the associated CCMPO. The CCMPO may specify, for example, that content in this particular channel may only be accessed by subscribers to the channel. A CMPS within the set-top box accesses a user profile persistently stored in NVRAM and determines that the user is a subscriber. The CMPS deems the CCMPO rule to have been satisfied.

4. The CMPS obtains an identifier for the MCMPO associated with the work (video) currently streaming on the channel and a key for the MCMPO. If works are received serially on the channel (e.g., a television channel in which one work is provided at a time), the received MCMPO identifier may include don't care bits so that it can address any MCMPO currently on the channel.

5. The CMPS begins demuxing of Aggregate Stream 2501 (this may occur in parallel with the preceding step), and obtains the MCMPO, which is encoded into an ES multiplexed within the Aggregate Stream (e.g., MCMPO 2506). Although each ES within Aggregate Stream 2501 has been encrypted, Aggregate Stream 2501 was not encrypted following multiplexing. This allows the CMPS to demultiplex Aggregate Stream 2501 without decrypting the entire Aggregate Stream.

6. The CMPS identifies the ES which constitutes the MCMPO (e.g., ES 2503). The CMPS downloads one complete instance of MCMPO 2506 into an internal buffer, and uses the key received from CCMPO 2504 to decrypt MCMPO 2506.

7. The CMPS determines which rules are applied by MCMPO 2506. MCMPO 2506 might, for example, include a rule stating that the user can view the associated work with advertisements at a low fee, but must pay a higher fee for viewing the work without advertisements.

8. The CMPS generates an options menu, and displays that menu on the screen for the user. The menu specifies the options, including the cost for each option. Additional options may be specified, including payment types.

9. The user uses a remote control pointing device to choose to view the work at a lower cost but with advertisements. The user specifies that payment can be made from an electronic cash budget stored in the CMPS.

10. The CMPS subtracts the specified amount from the budget persistently stored in NVRAM, and generates and encrypts a message to a server associated with the cable. The message transfers the required budget to the server, either by transferring electronic cash, or by authorizing a financial clearinghouse to transfer the amount from the user's account to the cable provider's. This message may be sent immediately, or may be buffered to be sent later (e.g., when the user connects the device to the Internet). This step may be taken in parallel with decryption of the content.)

11. The CMPS obtains from MCMPO 2506 a set of keys used to decrypt the Elementary Streams associated with the work (e.g., ES 2502). The CMPS also obtains identifiers for the specific ESs to be used. Since the user has indicated that advertisements are to be included, the MCMPO identifies ESs associated with the advertisements, and identifies a Scene Descriptor Graph which includes advertisements. A Scene Descriptor Graph which does not include advertisements is not identified, and is not passed through by the CMPS.

12. The CMPS passes the decrypted ESs to the MPEG-4 buffers. The normal process of MPEG-4 decoding, compositing and rendering then takes place. The Composite

and Render block outputs Start and Stop events for each object released for viewing. The CMPS monitors this information and compares it to the expected events. In particular, the CMPS confirms that the advertisements have been released for viewing, and that each operation has occupied approximately the expected amount of time.

In another embodiment, a set-top box containing a CMPS (e.g., CMPS 2302 from FIG. 23) may have a cable input (e.g., carrying M4 Bit Streams 2314 and CMPOs 2303). The cable may carry multiple channels, each made up of two sub-channels, with one sub-channel carrying MPEG-4 ESs (e.g., M4 Bit Streams 2314), and the other sub-channel carrying CMPOs (e.g., CMPOs 2303). The sub-channel carrying CMPOs 2303 could be routed directly to CMPS 2302, with the ES channel being routed to a decryption block (operating under control of the CMPS, e.g., CR&D 2315), and then to the MPEG-4 buffers (e.g., buffers associated with Scene Descriptor Graph 2306, AVO Decode 2307 and Object Descriptors 2308). In this case, if the ESs are not encrypted, they proceed unchanged through the decryption block and into the buffers. This may occur, for example, if the ESs are being broadcast for free, with no restrictions, and/or if they are public domain information, and/or they were created prior to inclusion of CMPOs in the MPEG-4 standard.

Such an embodiment might include timing synchronization information in the CMPO sub-channel, so that CMPOs can be synchronized with the associated ESs.

The concept of incorporating two separate streams, one consisting of control information and connected directly to the CMPS, and the other consisting of ESs, may support a high degree of modularization, such that the formats of CMPOs, and particular types of CMPS's, may be changed without alteration to the underlying ES format. For example, it may be possible to change the CMPO format without the necessity for reformatting content ESs. To take another example, it may be possible to upgrade a Commerce Appliance by including a new or different CMPS, without the necessity for any changes to any of the circuitry designed to demultiplex, composite and render the content ESs. A user might obtain a CMPS on a smart card or other removable device, and plug that device into a Commerce Appliance. This could be done to customize a Commerce Appliance for a particular application or for particular content.

CMPS Interface to a CE Device

A CMPS may be designed to present a standardized interface between the general-purpose functionality of a consumer electronics device and any relevant CMPOs and/or other CI and protected content. For example, a CMPS could be designed to accept CI and encrypted ESs, and output decrypted ESs into the device's buffers. In such a case, the manufacturer of the device would be able to design the device in compliance with the specification (e.g., MPEG-4), without concern about commerce-related extensions to the standard, which extensions might differ from provider to provider. All such extensions would be handled by the CMPS.

Initialization

1. Initialization of the CMPS.

A CMPS may be used to identify the capabilities of the Commerce Appliance in which a CMPS is installed. A CMPS permanently associated with a particular Commerce Appliance may have such information designed-in when the CMPS is initially installed (e.g., stored in ROM 2406 shown in FIG.24). A CMPS which is removable may be used to run an initialization operation in order to obtain information about the device's capabilities. Such information may be stored in a data structure stored in NVRAM 2425. Alternatively, some or all of such information may be gathered each time the device is turned on, and stored in RAM 2414.

For example, a DVD player may or may not contain a connection to an external server and/or process. A CMPO and/or other CI stored on a DVD (and/or any other format optical disk) inserted into a DVD (or any other format optical disk) player may include rules predicated on the possibility of outputting information to a server (e.g., content is free if user identification information is output), or may require a direct connection in order, for example, to download keys used to decrypt content. In such a case, the CMPS arrangement may determine the hardware functionality which is expected by or required by the CMPO, and compare that to the hardware actually present. If the CMPS determines that the CMPO and/or other CI requires a network connection , and that the DVD player does not include such a connection, the CMPS may take a variety of steps, including: (1) if the network connection is required for some options but not others, causing only those options which are possible to be displayed to the user; (2) informing the user that necessary hardware is missing; or (3) causing a graceful rejection of the disk, including informing the user of the reason for the rejection.

To take another example, a CMPO and/or other CI may include a business model which allows the user to choose among quality levels (or other forms of variations of a given work, for example, longer length and/or greater options), with a higher price being charged if the user selects a higher level of quality (e.g., music may be played at low resolution for free, but requires a payment in order to be played at a higher resolution). In such a case, the Commerce Appliance may not include loudspeakers which are capable of outputting sound at the higher resolution. The CMPS arrangement preferably identifies this situation, and either eliminates the higher resolution output as an option for the user, or informs the user that this option costs more but provides no additional benefit given the Commerce Appliance's current functionality or given the Commerce Appliance not being docked in a user arrangement that provides higher quality loudspeakers.

If the Commerce Appliance may be hooked up to external devices (e.g., loudspeakers, display, etc.), the CMPS will require some mechanism for identifying and registering such devices. Each device may be used to make standard ID and capability information available at all times, thereby allowing the CMPS to poll all connected devices at regular intervals, including, for example, authenticating CMPS arrangements within one or more of each such connected devices. Using another approach, all devices could be used to output CMPS identification information upon power-on, with later connected devices being used to output such information upon establishment of the connection. Such identification information may take the form, for example, of authentication information provided under the "five company arrangement", such authentication methods are herein incorporated by reference.

As discussed earlier, a Commerce Appliance may be connected to multiple devices each containing its own CMPS arrangement (e.g., a DVD player may be connected to a digital TV) In such cases, the CMPSs must be able to initiate secure communication (e. g., using a scheme, for example, like the "five company proposal" for IEEE 1394 serial bus) and determine how the CMPSs will interact with respect to content communication between CMPSs and, in certain embodiments, regarding cooperative governance of such content such as describing in the incorporated Shear patent application. In one embodiment, the first CMPS arrangement to receive content might govern the control process by downloading an initial CMPO and/or other CI, and display one or more of the rules to the user, etc. The second CMPS arrangement might recognize that it has no further role to play, either as a result of a communication between the two CMPS arrangements, or as a result of changes to the content stream created by the first CMPS arrangement (which decrypted the content, and may have allowed demuxing, composition and rendering, etc.)

The relationship between upstream and downstream CMPSs arrangements may be complicated if one device handles certain aspects of MPEG-4 rendering, and the other handles other aspects. For example, a DVD player might handle demuxing and buffering, transferring raw ESs to a digital TV, which then handles composition and rendering, as well as display. In such a case, there might be no back-channel from the composition and rendering block to the upstream CMPS arrangement. CMPS arrangements are preferably designed to handle stand-alone cases (a DVD (or any other optical disk) player with a CMPS arrangement attached to a dumb TV with no CMPS), multiple CMPS arrangement cases in which one CMPS arrangement handles all of the processing (a DVD (or other optical disk) player which handles everything through composition and rendering, with a video stream output to the digital TV (in one non-limiting example, via an IEEE 1349 serial bus) (that output stream would be encrypted as per the "five company proposal" for copy protection using IEEE 1394 serial bus transmission)) and/or shared processing between two or more CMPSs arrangements regarding some, or in certain cases, all, of such processing.

2. Initialization of a particular content stream.

The CMPS may be designed so that it can accept initialization information which initializes the CMPS for a particular content stream or channel. This header, which may be a CMPO and/or other CI, may contain information used by the CMPS to locate and/or interpret a particular content stream as well as CI associated with that stream. This initial header may be received through a sideband channel, or may be received as a CI ES such as a CMPO ES.

In one example, shown in FIG. 26, Header CMPO 2601 may include the following information:

(a) Stream/Object/CMPO ID 2602, which identifies the content streams/objects governed by Header CMPO 2601 and/or identification of CMPOs associated with each such content stream or object.

In one embodiment, Header CMPO 2601 identifies other CMPOs which contain rules and keys associated with particular content streams. In another embodiment, Header CMPO 2601 directly controls all content streams, by incorporating the keys and rules associated with such streams. In the latter case, no other CMPOs may be used.

In one embodiment, Header CMPO 2601 may be one or more CMPOs, CCMPOs, MCMPOs, and/or other CI.

(b) One or CMPO Keys 2603 for decrypting each identified CMPO.

(c) Work-Level Control 2604, consisting of basic control information associated with the work as a whole, and therefore potentially applicable to all of the content streams which make up the work. This basic control information may include rules governing the work as a whole, including options to be presented to the user.

(d) In one embodiment of this embodiment, a header CMPO may be updatable to contain User/Site Information 2605 regarding a particular user or site currently authorized to use certain content, as well as one or more rule sets under which the user has gained such authorization. A header CMPO associated with a work currently being viewed may be stored in RAM or NVRAM. This may include updated information. In one embodiment, the CMPO may also store header CMPOs for certain works viewed in the past. In one embodiment, header CMPOs may be stored in non-secure memory, with information sufficient to identify and authenticate that each header CMPO had not been changed.

In one such header CMPO embodiment of this embodiment, the header CMPO operates as follows:

(a) The header CMPO is received by a CMPS arrangement. In the case of previously unreceived content which has now become available, the header CMPO may be received at an input port. In the case of content which is already available, but is not currently being used (e.g., a set-top box with 500 channels, of which either 0 or 1 are being displayed at any given time), CCMPOs for each channel may be buffered by the CMPS arrangement for possible use if the user invokes particular content (e.g., switches to a particular channel).

In either case, the header CMPO must include information which allows a CMPS arrangement to identify it as a header CMPO.

(b) The CMPS arrangement obtains business-model information held in the clear in the header CMPO. Business-model information may include, for example, a statement that content can be viewed for free if advertisements are included, or if the user authorizes Nielson-type information, user and/or audience measurement information, for example, content may be output to a server or otherwise copied once, but only at a price.

(c) The CMPS arrangement either accepts the business model, if the user has authorized it to accept certain types of models (e.g., the user has programmed the CMPS arrangement to always accept play with advertisements for free), rejects the business model, if the user has instructed that the particular model always be rejected, or displays the business model to the user (e.g., by presenting options on the screen).

(d) If a business model has been accepted, the CMPS arrangement then decrypts the remainder of the header CMPO. If the Commerce Appliance contains a live output connection to an external server (e.g., Internet connection, back-channel on a set-top box, etc.), and if latency problems are handled, decryption of these keys can be handled by communicating with the external server, each side authenticating the other, establishment of a secure channel, and receipt of a key from the server. If the Commerce Appliance is not at least occasionally connected to an external server, decryption may have to be based on one or more keys securely stored in the Commerce Appliance.

(e) Once a header CMPO has been decrypted, the CMPS arrangement acquires information used to identify and locate the streams containing the content, and keys which are used to decrypt either the CMPOs associated with the content, or to directly decrypt the content itself.

(f) In one embodiment of this header embodiment, the header CMPO may contain a data structure for the storage of information added by the CMPS arrangement. Such information may include the following:

(1) Identification of user and/or Commerce Appliance and/or CMPS arrangement. In this embodiment, such information may be stored in a header CMPO in order to provide an audit trail in the event the work (including the header CMPO) is transferred (this only works if the header CMPO is transferred in a writable form). Such information may be used to allow a user to transfer the work to other Commerce Appliances owned by the user without the payment of additional cost, if such transfers are allowed by rule information associated with the header CMPO. For example, a user may have a subscription to a particular cable service, paid for in advance by the user. When a CMPS arrangement downloads a header CMPO from that cable service, the CMPS arrangement may store the user's identification in the header CMPO. The CMPS arrangement may then require that the updated header CMPO be included if the content is copied or transferred. The header CMPO could include a rule stating that, once the user information has been filled in, the associated content can only be viewed by that user, and/or by Commerce Appliances associated with that user. This would allow the user to make multiple copies of the work, and to display the work on multiple Commerce Appliances, but those copies could not be displayed or used by non-authorized users and/or on non-authorized Commerce Appliances. The header CMPO might also include a rule stating that the user information can only be changed by an authorized user (e.g., if user 1 transfers the work to user 2, user 2's CMPS arrangement can update the user information in the header CMPO, thereby allowing user 2 to view the work, but only if user 2 is also a subscriber to the cable channel).

(2) Identification of particular rules options governing use. Rule sets included in header CMPOs may include options. In certain cases, exercise of a particular option might preclude later exercise of a different option. For example, a user might be given the choice to view an unchanged work for one price, or to change a work and view the changed work for a higher price. Once the user decides to change the work and view the changed work, this choice is preferably stored in the header CMPO, since the option of viewing the original unchanged work at the lower price is no longer available. The user might have further acquired the right, or may now be presented with the option for the right, to further distribute the changed work at a mark-up in cost resulting in third party

derived revenue and usage information flowing to both the user and the original work stakeholder(s).

(3) Historical usage information. The header CMPO may include information relating to the number and types of usages. For example, if the underlying work is copied, the header CMPO may be updated to reflect the fact that a copy has been made, since a rule associated with the work might allow only a single copy (e.g., for backup and/or timeshifting purposes). To take another example, a user might obtain the right to view a work one time, or for a certain number of times. The header CMPO would then be updated to reflect each such use.

Usage information may be used to determine if additional uses are authorized by rules associated with the header CMPO. Such information may also be used for audit purposes. Such information may also be provided as usage information exhaust, reported to an external server. For example, a rule may specify that a work may be viewed for free, but only if historical usage information is downloaded to a server.

Content Management Protection Objects (CMPO)

The Content Management and Protection Object ("CMPO") is a data structure which includes information used by the CMPS to govern use of certain content. A CMPO may be formatted as a data structure specified by a particular standard (e.g., an MPEG-4 ES), or may be formatted as a data structure not defined by the standard. If the CMPO is formatted as a data structure specified by the standard, it may be received in the channel utilized by the standard (e.g., as part of a composite MPEG-4 stream) or may be received through some other, side-band method. If the CMPO is formatted as a data structure not specified by the relevant standard, it is provided and decoded using some side-band method, which may include receipt through the same port as formatted content and/or may include receipt through a separate port.

Content may be controlled at virtually any level of granularity. Three exemplary levels will be discussed herein: "channel," "work," and "object."

A "channel" represents an aggregation of works. The works may be available for selection by the user (e.g., a web site, or a video library) or may be received serially (e.g., a cable television channel).

A "work" represents a single audio-visual, textual or other work, intended to be consumed (viewed, read, etc.) by a user as an integrated whole. A work may, for example, be a movie, a song, a magazine article, a multimedia product such, for example, as sophisticated videogame. A work may incorporate other works, as, for example, in a multimedia work which incorporates songs, video, text, etc. In such a case, rights may be associated

An "object" represents a separately addressable portion of a work. An object may be, for example, an individual MPEG-4 AVO, a scene descriptor graph, an object descriptor, the soundtrack for a movie, a weapon in a videogame, or any other logically definable portion.

Content may be controlled at any of these levels (as well as intermediate levels not discussed herein). The preferred embodiment mechanism for such control is a CMPO or CMPO arrangement (which comprises one or more CMPOs, and if plural, then plural, cooperating CMPOs). CMPOs and CMPO arrangements may be organized hierarchically, with a Channel CMPO arrangement imposing rules applicable to all contained works, a MCMPO or an SGCMPO imposing rules applicable to all objects within a work, and a CMPO arrangement imposing rules applicable to a particular object.

In one embodiment, illustrated in FIG. 27, a CMPS may download CCMPO 2701. CCMPO 2701 may include one or more Rules 2702 applicable to all content in the channel, as well as one or more Keys 2703 used for decryption of one or more MCMPOs and/or SGCMPOs. MCMPO 2704 may include Rules 2705 applicable to a single work and/or works, one or more classes and/or more users and/or user classes, and may also include Keys 2706 used to decrypt CMPOs. CMPO 2707 may include Rules 2708 applicable to an individual object, as well as Key 2709 used to decrypt the object.

As long as all objects are subject to control at some level, there is no requirement that each object be individually controlled. For example, CCMPO 2701 could specify a single rule for viewing content contained in its channel (e.g., content can only be viewed by a subscriber, who is then might be free to redistribute the content with no further obligation to the content provider). In such a case, rules would not necessarily be used for MCMPOs (e.g. Rules 2705), SGCMPOs, or CMPOs (e.g., Rules 2708). In one embodiment, MCMPOs, SGCMPOs, and CMPOs could be dispensed with, and CCMPO 2701 could include all keys used to decrypt all content, or could specify a location where such keys could be located. In another embodiment, CCMPO 2701 would supply Key 2703 used to decrypt MCMPO 2704. MCMPO 2704 might include keys used to decrypt CMPOs (e.g., Keys 2706), but might include no additional Rules 2705. CMPO 2707 might include Key 2709 used to decrypt an object, but might include no additional Rules 2708. In certain embodiments, there may be no SGCMPOs.

A CMPO may be contained within a content data structure specified by a relevant standard (e.g., the CMPO may be part of a header in an MPEG-4 ES.) A CMPO may be contained within its own, dedicated data structure specified by a relevant standard (e.g., a

CMPO ES). A CMPO may be contained within a data structure not specified by any content standard (e.g., a CMPO contained within a DigiBox).

A CCMPO may include the following elements:

(a) ID 2710. This may take the following form: <channel ID>< CMPO type><CMPO ID><version number>. In the case of hierarchical CMPO organization (e.g., CCMPOs controlling MCMPOs controlling CMPOs), CMPO ID 2711 can include one field for each level of the hierarchy, thereby allowing CMPO ID 2711 to specify the location of any particular CMPO in the organization. ID 2710 for a CCMPO may, for example, be 123-000-000. ID 2712 for a MCMPO of a work within that channel may, for example, be 123-456-000, thereby allowing the specification of 1,000 MCMPOs as controlled by the CCMPO identified as "123." CMPO ID 2711 for a CMPO associated with an object within the particular work may, for example, be 123-456-789, thereby allowing the specification of 1,000 CMPOs as associated with each MCMPO.

This method of specifying CMPO IDs thereby conveys the exact location of any CMPO within a hierarchy of CMPOs. For cases in which higher levels of the hierarchy do not exist (e.g., a MCMPO with no associated CCMPO), the digits associated with that level of the hierarchy may be specified as zeroes.

(b) Rules 2702 applicable to all content in the channel. These may be self-contained rules, or may be pointers to rules obtainable elsewhere. Rules are optional at this level.

(c) Information 2713 designed for display in the event the user is unable to comply with the rules (e.g., an advertisement screen informing the user that a subscription is available at a certain cost, and including a list of content available on the channel).

(d) Keys 2703 for the decryption of each MCMPO controlled by this CCMPO. In one embodiment, the CCMPO includes one or more keys which decrypt all MCMPOs. In an alternate embodiment, the CCMPO includes one or more specific keys for each MCMPO.

(e) A specification of a CMPS Type (2714), or of hardware/software necessary or desirable to use the content associated with this channel.

The contents of a MCMPO may be similar to those of a CCMPO, except that the MCMPO may include rules applicable to a single work, and may identify CMPOs associated with each object.

The contents of each CMPO may be similar to those of the MCMPO, except that the CMPO may include rules and keys applicable to a single object.

The contents of an SGCMPO may be similar to those of the CCMPO, except that the MCMPO may include rules applicable to only certain one or more classes of rights, certain one or more classes of works, and/or to one or more certain classes of users and/or user arrangements (e.g. CMPO arrangements and/or their devices).

In another embodiment, shown in FIG. 28, CMPO Data Structure 2801 may be defined as follows:

CMPO Data Structure 2801 is made up of elements. Each element includes a self-contained item of information. The CMPS parses CMPO Data Structure, one element at a time.

Type Element 2802 identifies the data structure as a CMPO, thereby allowing the CMPS to distinguish it from a content ES. In an exemplary embodiment, this element may include 4 bits, each of which may be set to "1" to indicate that the data structure is a CMPO.

The second element is CMPO Identifier 2803, which is used to identify this particular CMPO and to convey whether the CMPO is part of a hierarchical organization of CMPOs and, if so, where this CMPO fits into that organization.

CMPO Identifier 2803 is divided into four sub-elements, each of three bits. These are shown as sub-elements A, B, C and D. The first sub-element (2803 A) identifies the CMPO type, and indicates whether the CMPO is governed or controlled by any other CMPO:

100: this is a top-level CMPO (associated with a channel or an aggregation of works) and is not controlled by any other CMPO.

010: this is a mid-level CMPO (associated with a particular work) and is not controlled by any other CMPO.

110: this is a mid-level CMPO, and is controlled by a top-level CMPO.

001: this is a low-level CMPO (associated with an object within a work) and is not controlled by any other CMPO. This case will be rare, since a low-level CMPO will ordinarily be controlled by at least one higher-level CMPO.

011: this is a low-level CMPO, and is controlled by a mid-level CMPO, but not by a top-level CMPO.

111: this is a low-level CMPO, and is controlled by a top-level CMPO and by a mid-level CMPO.

The second sub-element of CMPO ID 2803 (sub-element B) identifies a top-level CMPO. In the case of a top-level CMPO, this identifier is assigned by the creator of the CMPO. In the case of a mid-level or low-level CMPO which is controlled by a top-level

CMPO, this sub-element contains the identification of the top-level CMPO which performs such control. In the case of a mid-level or low-level CMPO which is not controlled by a top-level CMPO, this sub-element contains zeroes.

The third sub-element of CMPO ID 2803 (sub-element C) identifies a mid-level CMPO. In the case of a top-level CMPO, this sub-element contains zeroes. In the case of a mid-level CMPO, this sub-element contains the identification of the particular CMPO. In the case of a low-level CMPO which is controlled by a mid-level CMPO, this sub-element contains the identification of the mid-level CMPO which performs such control. In the case of a low-level CMPO which is not controlled by a mid-level CMPO, this sub-element contains zeroes.

The fourth sub-element of CMPO ID 2803 (sub-element D) identifies a low-level CMPO. In the case of a top-level or mid-level CMPO, this sub-element contains zeroes. In the case of a low-level CMPO, this sub-element contains the identification of the particular CMPO.

Following the identifier element is Size Element 2804 indicating the size of the CMPO data structure. This element contains the number of elements (or bytes) to the final element in the data structure. This element may be rewritten if alterations are made to the CMPO. The CMPS may use this size information to determine whether the element has been altered without permission, since such an alteration might result in a different size. For such purposes, the CMPS may store the information contained in this element in a protected database. This information can also be used to establish that the entire CMPO has been received and is available, prior to any attempt to proceed with processing.

Following Size Element 2804 are one or more Ownership/Control Elements containing ownership and chain of control information (e.g., Ownership/Control Elements 2805, 2806 and 2807). In the first such element (2805), the creator of the CMPO may include a specific identifier associated with that creator. Additional participants may also be identified in following elements (e.g., 2806, 2807). For example, Element 2805 could identify the creator of the CMPO, Element 2806 could identify the publisher of the associated work and Element 2807 could identify the author of the work.

A specific End Element 2808 sequence (e.g., 0000) indicates the end of the chain of ownership elements. If this sequence is encountered in the first element, this indicates that no chain of ownership information is present.

Chain of ownership information can be added, if rules associated with CMPO 2801 permit such additions. If, for example, a user purchases the work associated with CMPO 2801, the user's identification may be added as a new element in the chain of ownership elements (e.g., a new element following 2807, but before 2808). This may be done at the point of purchase, or may be accomplished by the CMPS once CMPO 2801 is encountered and the CMPS determines that the user has purchased the associated work. In such a case, the CMPS may obtain the user identifier from a data structure stored by the CMPS in NVRAM.

Following the ownership element chain are one or more Handling Elements (e.g., 2809, 2810) indicating chain of handling. These elements may contain the identification of any CMPS which has downloaded and decoded CMPO 2801, and/or may contain the identification of any user associated with any such CMPS. Such information may be used for audit purposes, to allow a trail of handling in the event a work is determined to have been circulated improperly. Such information may also be reported as exhaust to a clearinghouse or central server. Chain of handling information preferably remains persistent until reported. If the number of elements required for such information exceeds a specified amount (e.g., twenty separate user identifiers), a CMPS may refuse to allow any further processing of CMPO 2801 or the associated work until the CMPS has been connected to an external server and has reported the chain of handling information.

The last element in the chain of handling elements (e.g., 2811) indicates the end of this group of elements. The contents of this element may, for example, be all zeroes.

Following the chain of handling elements may be one or more Certificate Elements (e.g., 2812, 2813) containing or pointing to a digital certificate associated with this CMPO. Such a digital certificate may be used by the CMPS to authenticate the CMPO. The final element in the digital certificate chain is all zeroes (2814). If no digital certificate is present, a single element of all zeroes exists in this location.

Following the Certificate Elements may be a set of Governed Object Elements (e.g., 2815, 2816, 2817, 2818) specifying one or more content objects and/or CMPOs which may be governed by or associated with CMPO 2801. Each such governed object or CMPO is identified by a specific identifier and/or by a location where such object or CMPO may be found (e.g., these may be stored in locations 2815 and 2817). Following each such identifier may be one or more keys used to decrypt such CMPO or object (e.g., stored in locations 2816 and 2818). The set of identifiers/keys ends with a termination element made up of all zeroes (2819).

Following the set of elements specifying identifiers and/or keys may be a set of Rules Elements (e.g., 2820, 2821, 2822) specifying rules/controls and conditions associated with use of the content objects and/or CMPOs identified in the Governed Objects chain (e.g., locations 2815 and 2817). Exemplary rules are described below. Elements may contain explicit rules or may contain pointers to rules stored elsewhere. Conditions may include particular hardware resources necessary to use associated content objects or to satisfy certain rules, or particular types of CMPS's which are necessary or preferred for use of the associated content objects.

Following the rules/controls and conditions elements may be a set of Information Elements 2823 containing information specified by the creator of the CMPO. Among other contents, such information may include content, or pointers to content, programming, or pointers to programming.

The CMPO ends with Final Termination Element 2824.

In one embodiment, the rules contained in Rules Elements 2820-2822 of CMPO 2801 may include, for example, the following operations:

(1) Play. This operation allows the user to play the content (though not to copy it) without restriction.

(2) Navigate. This allows the user to perform certain types of navigation functions, including fast forward/rewind, stop and search. Search may be indexed or unindexed.

(3) Copy. Copy may be allowed once (e.g., time-shifting, archiving), may be allowed for a specified number of times and/or may be allowed for limited period of time, or may be allowed for an unlimited period of time, so long as other rules, including relevant budgets, are not violated or exceeded. A CMPS arrangement may be designed so that a Copy operation may cause an update to an associated CMPO (e.g., including an indication that the associated content has been copied, identifying the date of copying and the site responsible for making the copy), without causing any change to any applicable content object, and in particular without requiring that associated content objects be demuxed, decrypted or decompressed. In the case of MPEG-4, for example, this may require the following multi-stage demux process:

(i) the CMPS arrangement receives a Copy instruction from the user, or from a header CMPO.

(ii) CMPO ESs associated with the MPEG-4 stream which is to be copied are separated from the content stream in a first demux stage.

(iii) CMPOs are decrypted and updated by the CMPS arrangement. The CMPOs are then remuxed with the content ESs (which have never been demuxed from each other), and the entire stream is routed to the output port without further alteration.

This process allows a copy operation to take place without requiring that the content streams be demuxed and decrypted. It requires that the CMPS arrangement include two outputs: one output connected to the digital output port (e.g., FIG. 23 line 2316, connecting to Digital Output Port 2317), and one output connected to the MPEG-4 buffers (e.g., FIG. 23, lines 2310, 2311, 2312), with a switch designed to send content to one output or the other (or to both, if content is to be viewed and copied simultaneously) (e.g., Switch 2319). Switch 2319 can be the only path to Digital Output Port 2317, thereby allowing CMPS 2302 to exercise direct control over that port, and to ensure that content is never sent to that port unless authorized by a control. If Digital Output Port 2317 is also the connector to a digital display device, CMPS 2302 will also have to authorize content to be sent to that port even if no copy operation has been authorized.

In one example embodiment, the receiving device receiving the information through Digital Output Port 2317 may have to authenticate with the sending device (e.g., CMPS 2302). Authentication may be for any characteristic of the device and/or one or more CMPSs used in conjunction with that device. Thus, for example, a sending appliance may not transmit content to a storage device lacking a compatible CMPS.

In another non-limiting example, CMPS 2302 can incorporate session encryption functionality (e.g., the "five company arrangement") which establishes a secure channel from a sending interface to one or more external device interfaces (e.g., a digital monitor), and provided that the receiving interface has authenticated with the sending interface, encrypts the content so that it can only be decrypted by one or more authenticated 1394 device interfaces. In that case, CMPS 2302 would check for a suitable IEEE 1394 serial bus interface , and would allow content to flow to Digital Output Port 2317 only if (a) an authorized Play operation has been invoked, a secure channel has been established with the device and the content has been session-encrypted, or (b) an authorized Copy or Retransmit operation has been invoked, and the content has been treated as per the above description (i.e., the CMPO has been demuxed, changed and remuxed, the content has never been decrypted or demuxed).

This is only possible if CMPOs are separately identifiable at an early demux stage, which most likely requires that they be stored in separate CMPO ESs. If the CMPOs are stored as headers in content ESs, it may be impossible to identify the CMPOs prior to a full demux and decrypt operation on the entirety of the stream.

(4) Change. The user may be authorized to change the content.

(5) Delete. This command allows the user to delete content which is stored in the memory of the Consumer Appliance. This operation operates on the entire work. If the user wishes to delete a portion of a work, the Change operation must be used.

(6) Transfer. A user may be authorized to transfer a work to a third party.

This differs from the Copy operation in that the user does not retain the content or any rights to the content. The Transfer operation may be carried out by combining a Copy operation and a Delete operation. Transfer may require alteration of the header CMPO associated with the work (e.g., adding or altering an Ownership/Control Element, such as Elements 2805-2807 of FIG. 28), so as to associate rights to the work with the third party.

These basic operations may be subject to modifications, which may include:

i. Payment. Operations may be conditioned on some type of user payment. Payment can take the form of cash payment to a provider (e.g., credit card, subtraction from a budget), or sending specified information to an external site (e.g., Nielson-type information).

ii. Quality of Service. Operations may specify particular quality of service parameters (e.g., by specifying a requested QoS in MPEG-4), including: requested level of decompression, requested/required types of display, rendering devices (e.g., higher quality loudspeakers, a particular type of game controller).

iii. Time. Operations may be conditioned such that the operation is only allowed after a particular time, or such that the price for the operation is tied to the time (e.g., real-time information at a price, delayed information at a lower price or free, e.g., allowing controlled copies but only after a particular date).

iv. Display of particular types of content. Operations may be conditioned on the user authorizing display of certain content (e.g., the play operation may be free if the user agrees to allow advertisements to be displayed).

In all of these cases, a rule may be modified by one or more other rules. A rule may specify that it can be modified by other rules or may specify that it is unmodifiable. If a rule is modifiable, it may be modified by rules sent from other sources. Those rules may be received separately by the user or may be aggregated and received together by the user.

Data types which may be used in an exemplary MPEG-4 embodiment may include the following:

a. CMP Data Stream.

The CMP-ds is a new elementary stream type that has all of the properties of an elementary stream including its own CMPO and a reference in the object descriptors. Each CMP-ds stream has a series of one or more *CMP Messages*. A CMP_Message has four parts:

1. Count: [1...*n*] CMPS types supported by this IP ES. Multiple CMPS systems may be supported, each identified by a unique *type*. (There may have to be a central registry of types.)

2. **CMPS_type_identifiers:** [1...*n*] identifiers, each with an offset in the stream and a length. The offset points to the byte in the CMPO where the data for that CMPS type is found. The length is the length in bytes of this data.

3. **Data segments**: One segment for each of the *n* CMPS types encoded in a format that is proprietary to the CMPS supplier.

4. **CMP_Message_URL:** That references another CMP_Message. (This is in keeping with the standard of using URLs to point to streams.)

b. CMPO.

The CMPO is a data structure used to attach detailed CMP control to individual elementary streams. Each **CMPO** contains:

1. **CMPO_ID:** An identifier for the content under control. This identifier must *uniquely* identify an elementary stream.

2. **CMPO_count:** [1...*n*] CMPS types supported by this **CMPO**.

3. **CMPS_type_identifiers:** [1...*n*] identifiers, each with an offset in the stream and a length. The offset points to the byte in the CMPO where the data for that CMPS type is found. The length is the length in bytes of this data.

4. **Data segments:** *n* data segments. Each data segment is in a format that is proprietary to the CMPS supplier.

5. **CMPO_URL:** An optional URL that references an additional CMPO that adds information to the information in this CMPO. (This is a way of dynamically adding support for new CMPSs.)

c. Feedback Event

The feedback events come in two forms: start and end. Each feedback event contains three pieces of information:

1. **Elementary_stream_ID**

2. **Time:** in presentation time

3. **Object_instance_number**

User Interface.

Commerce Appliance 2301 may include User Interface 2304 designed to convey control-related information to the user and to receive commands and information from the user. This interface may include special purpose displays (e.g., a light which comes on if a current action requires payment), special purpose buttons (e.g., a button which accepts the payment or other terms required for display of content), and/or visual information presented on screen.

Example of Operation in an MPEG-4 Context

1. User selects a particular work or channel. The user may, for example, use a remote control device to tune a digital TV to a particular channel.

2. Selection of the channel is communicated to a CMPS arrangement, which uses the information to either download a CCMPO or to identify a previously downloaded CCMPO (e.g., if the CMPS arrangement is contained in a set-top box, the set-top box may automatically download CCMPOs for every channel potentially reachable by the box).

3. The CMPS arrangement uses the CCMPO to identify rules associated with all content found on the channel. For example, the CCMPO may specify that content may only be viewed by subscribers, and may specify that, if the user is not a subscriber, an advertisement screen should be put up inviting the user to subscribe.

4. Once rules specified by the CCMPO have been satisfied, the CCMPO specifies the location of a MCMPO associated with a particular work which is available on the channel. The channel CMPO may also supply one or more keys used for decryption of the MCMPO.

5. The CMPS arrangement downloads the MCMPO. In the case of an MPEG-4 embodiment, the MCMPO may be an Elementary Stream. This Elementary Stream must be identifiable at a relatively early stage in the MPEG-4 decoding process.

6. The CMPS arrangement decrypts the MCMPO, and determines the rules used to access and use the content. The CMPS arrangement presents the user with a set of options, including the ability to view for free with advertisements, or to view for a price without advertisements.

7. The user selects view for free with advertisements, e.g., by highlighting and selecting an option on the screen using a remote control device.

8. The CMPS arrangement acquires one or more keys from the MCMPO and uses those keys to decrypt the ESs associated with the video. The CMPS arrangement identifies two possible scene descriptor graphs, one with and one without advertisements. The CMPS arrangement passes the scene descriptor graph with advertisements through, and blocks the other scene descriptor graph.

9. The CMPS arrangement monitors the composite and render block, and checks to determine that the advertisement AVOs have actually been released for viewing. If the CMPS arrangement determines that those AVOs have not been released for viewing, it puts up an error or warning message, and terminates further decryption.

CMPS Rights Management In Provider And Distribution Chains

In addition to consumer arrangements, in other embodiments one or more CMPSs may be used in creating, capturing, modifying, augmenting, animating, editing, excerpting, extracting, embedding, enhancing, correcting, fingerprinting, watermarking, and/or rendering digital information to associate rules with digital information and to enforce those rules throughout creation, production, distribution, display and/or performance processes.

In one non-limiting example, a CMPS, a non-exhaustive example of which may include a least a secure portion of a VDE node as described in the aforementioned Ginter et al., patent specification, is incorporated in video and digital cameras, audio microphones, recording, playback, editing, and/or noise reduction devices and/or any other digital device. Images, video, and/or audio, or any other relevant digital information may be captured, recorded, and persistently protected using at least one CMPS and/or at least one CMPO. CMPSs may interact with compression/decompression, encryption/decryption, DSP, digital to analog, analog to digital, and communications hardware and/or software components of these devices as well.

In another non-exhaustive example, computer animation, special effects, digital editing, color correcting, noise reduction, and any other applications that create and/or use digital information may protect and/or manage rights associated with digital information using at least one CMPS and/or at least one CMPO.

Another example includes the use of CMPSs and/or CMPOs to manage digital assets in at least one digital library, asset store, film and/or audio libraries, digital vaults, and/or any other digital content storage and management means.

In accordance with the present applications, CMPSs and/or CMPOs may be used to manage rights in conjunction with the public display and/or performance of digital works. In one non-exhaustive example, flat panel screens, displays, monitors, TV projectors, LCD projectors, and/or any other means of displaying digital information, may incorporate at least one hardware and/or software CMPS instance that controls the use of digital works. A CMPS may allow use only in conjunction with one or more digital credentials, one example of which is a digital certificate, that warrant that use of the digital information will occur in a setting, location, and/or other context for public display and/or performance. Non-limiting examples of said contexts include theaters, bars, clubs, electronic billboards, electronic displays in public areas, or TVs in airplanes, ships, trains and/or other public conveyances. These credentials may be issued by trusted third parties such as certifying authorities, non-exhaustive examples of which are disclosed in the aforementioned Ginter '712 patent application.

Additional MPEG-4 Embodiment Information

This work is based on the MPEG-4 description in the version 1 Systems Committee Draft (CD), currently the most complete description of the evolving MPEG-4 standard.

This section presents the structural modifications to the MPEG-4 player architecture and discusses the data lines and the concomitant functional changes. Figure 23 shows the functional components of the original MPEG-4 player. Content arrives at Player 2301 packaged into a serial stream (e.g., MPEG-4 Bit Stream 2314). It is demultiplexed via a sequence of three demultiplexing stages (e.g., Demux 2305) into elementary streams. There are three principle types of elementary streams: AV Objects (AVO), Scene Descriptor Graph (SDG), and Object Descriptor (OD). These streams are fed into respective processing elements (e.g., AVO Decode 2307, Scene Descriptor Graph 2306, Object Descriptors 2308). The AVOs are the multimedia content streams such as audio, video, synthetic graphics and so on. They are processed by the player's compression/coding subsystems. The scene descriptor graph stream is used to build the scene descriptor graph. This tells Composite and Render 2309 how to construct the scene and can be thought of as the "script." The object descriptors contain description information about the AVOs and the SD-graph updates.

To accommodate a CMPS (e.g., CMPS 2302) and to protect content effectively, the player structure must be modified in several ways:

- Certain data paths must be rerouted to and from the CMPS
- Certain buffers in the SDG, AVO decode and Object descriptor modules must be secured
- Feedback paths from the user and the composite and render units to the CMPS must be added

In order for CMPS 2302 to communicate with the MPEG-4 unit, and for it to effectively manage content we must specify the CMPO structure and association protocols and we must define the communication protocols over the feedback systems (from the compositor and the user.)

The structural modifications to the player are shown in Figure 23. The principal changes are:

- All elementary streams are now routed through CMPS 2302.
- Direct communication path between Demux 2305 and CMPS 2302.
- A required "Content Release and Decrypt" Module 2315 in CMPS 2302.

- The addition of a feedback loop (e.g., Line 2313) from Composite and Render 2309 to CMPS 2302.
- Bi-directional user interaction directly with the CMPS 2302, through Line 2316.

Furthermore, for M4v2P, CMP-objects are preferably associated with all elementary streams. Elementary streams that the author chooses not to protect are still marked by an "unprotected content" CMPO. The CMPOs are the primary means of attaching rules information to the content. Content here not only refers to AVOs, but also to the scene descriptor graph. Scene Descriptor Graph may have great value and will thus need to be protected and managed by CMPS 2302.

The direct path from Demux 2305 to CMPS 2302 is used to pass a CMPS specific header, that potentially contains business model information, that communicates business model information at the beginning of user session. This header can be used to initiate user identification and authentication, communicate rules and consequences, and initiate up-front interaction with the rules (selection of quality-of-service (QoS), billing, etc.) The user's communication with CMPS 2302 is conducted through a *non-standardized* channel (e.g., Line 2316). The CMPS designer may provide an independent API for framing these interactions.

Feedback Path 2313 from Composite and Render block 2309 serves an important purpose. The path is used to cross check that the system actually presented the user with a given scene. Elementary streams that are processed by their respective modules may not necessarily be presented to the user. Furthermore, there are several fraud scenarios wherein an attacker could pay once and view multiple times. The feedback path here allows CMPS 2302 to cross check the rendering and thereby perform a more accurate accounting. This feedback is implemented by forcing the Composite and Render block 2309 to issue a *start event* that signals the initiation of a given object's rendering that is complemented by a *stop event* upon termination. The feedback signaling process may be made optional by providing a CMP-notification flag that may be toggled to indicate whether or not CMPS 2302 should be notified. All CMPOs would be required to carry this flag.

The final modification to the structure is to require that the clear text buffers in the AVO, SDG and Object Descriptor processors and in the Composite-and-Render block be secured. This is to prevent a pirate from stealing content in these buffers. As a practical matter, this may be difficult, since tampering with these structures may well destroy synchronization of the streams. However, a higher state of security would come from placing these buffers into a protected processing environment.

CMPS 2302 *governs* the functioning of Player 2301, consistent with the following:

- Communication mechanism between CMPS 2302 and the MPEG-4 player (via CMPOs)
- A content release and decryption subsystem
- Version authentication subsystem
- Sufficient performance so as not to interfere with the stream processing in the MPEG-4 components

CMPS 2302 may have a bi-directional side-channel that is external to the MPEG-4 player that may also be used for the exchange of CMP information. Furthermore, the CMPS designer may choose to provide a user interface API that provides the user with the ability to communicate with the content and rights management side of the stream management (e.g., through Line 2316).

Encrypted content is decrypted and released by CMPS 2302 as a function of the rules associated with the protected content and the results of user interaction with CMPS 2302. Unencrypted content is passed through CMPS 2302 and is governed by associated rules and user interaction with CMPS 2302. As a consequence of these rules and user interaction, CMPS 2302 may need to transact with the SDG and AVO coding modules (e.g., 2310, 2311) to change scene structure and/or the QoS grade.

Ultimately, the CMPS designer may choose to have CMPS 2302 generate audit trail information that may be sent to a clearinghouse authority via CMPS Side Channel Port 2318 or as encrypted content that is packaged in the MPEG-4 bit stream.

The MPEG-4 v1 Systems CD uses the term "object" loosely. In this document, "object" is used to specifically mean a data structure that flows from one or more of the data paths in Figure 23.

Using multiple SD-graph update streams, each with its own CMPO, allows an author to apply arbitrarily specific controls to the SD-graph. For example, each node in the SD-graph can be created or modified by a separate SD-graph update stream. Each of these streams will have a distinct CMPO and ID. Thus, the CMPS can release and decrypt the creation and modification of each node and receive feedback information for each node individually. The practical implications for controlling release and implementing consequences should be comparable to having a CMPO on each node of the SD-graph, without the costs of having a CMPO on each SD-graph node.

Principles consistent with the present invention may be illustrated using the following examples:

In the first example, there is a bilingual video with either an English or French soundtrack. The user can choose during playback to hear either the English or French. The basic presentation costs $1. If the French soundtrack is presented there is a $0.50 surcharge. If the user switches back and forth between French and English, during a single viewing of the presentation, the $0.50 surcharge will occur only once.

In this example, there will be four elementary streams:

The Scene Description Graph Update stream will have a CMPO. The CMPO will imply a $1.00 fee associated with the use of the content. The scene description graph displays the video, English audio and puts up a button that allows the user to switch to French. If the user clicks that button, the English stops, the French picks up from that point and the button changes to a switch-to-English button. (Optionally, there may be a little dialog at the beginning to allow the user to select the initial language. This is all easy to do in the SD graph.)

The Video Stream with the CMPO will say that it can only be released if the scene description graph update stream above is released.

The English Audio Stream will be similar to the Video stream.

The French Audio Stream will be similar to the Video stream but there is a $.50 charge it if is seen in the feedback channel. (The CMPS must to not count twice if the user switches between the two in a single play of the presentation.)

An important requirement is that the ID for the SD-graph update stream appears in the feedback path (e.g., Feedback Path 2313). This is so CMPS 2302 knows when the presentation stops and ends so that CMPS 2302 can correctly bill for the French audio.

The rules governing the release of the video and audio streams may include some variations. The rules for these streams, for example, may state something like "if you don't see the id for the scene description graph update stream X in the feedback channel, halt release of this stream." If the main presentation is not on the display, then the video should not be. This ties the video to this one presentation. Using the video in some other presentation would require access to the original video, not just this protected version of it.

In a second example, an author wants to have a presentation with a free attract sequence or "trailer". If the user clicks the correct button the system moves into the for-fee presentation, which is organized as a set of "acts".

Multiple SD-graph update streams may update a scene description graph. Multiple SD-graph update streams may be open in parallel. The time stamps on the ALUs in the streams are used to synchronize and coordinate.

The trailer and each act are represented by a separate SD-graph update stream with a separate CMPO. There is likely an additional SD-graph update stream that creates a simple root node that is invisible and silent. This node brings in the other components of the presentation as needed.

The foregoing description of implementations of the invention has been presented for purposes of illustration and description. It is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but the present invention may be implemented as a combination of hardware and software or in hardware alone. The invention may be implemented with both object-oriented and non-object-oriented programming systems. The scope of the invention is defined by the claims and their equivalents.

We claim:

1. A streaming media player providing content protection and digital rights management, including:

a port configured to receive a digital bit stream, the digital bit stream including:

content which is encrypted at least in part, and

a secure container including control information for controlling use of the content, including at least one key suitable for decryption of at least a portion of the content; and

a control arrangement including:

means for opening secure containers and extracting cryptographic keys, and

means for decrypting the encrypted portion of the content.

2. The player of Claim 1 in which the digital bit stream includes at least two sub-streams which have been muxed together, at least one of the sub-streams including compressed information, and

wherein the player further includes:

a demux designed to separate and route the sub-streams;

a decompression unit configured to decompress at least one of the sub-streams, the decompression unit and the demux being connected by a pathway for the transmission of information; and

a rendering unit designed to process decompressed content information for rendering.

3. The player of Claim 2, further including:

a stream controller operatively connected to the decompression unit, the stream controller including decryption functionality configured to decrypt at least a portion of a sub-stream and pass the decrypted sub-stream to the decompression unit.

4. The player of Claim 3, further including:

a path between the control arrangement and the stream controller to enable the control arrangement to pass at least one key to the stream controller for use with the stream controller's decryption functionality.

5. The player of Claim 4, further including:

a feedback path from the rendering unit to the control arrangement to allow the control arrangement to receive information from the rendering unit regarding the identification of objects which are to be rendered or have been rendered.

6. The player of Claim 1, wherein the digital bit stream is encoded in MPEG-4 format.

7. The player of Claim 1, wherein the digital bit stream is encoded in MP3 format.

8. The player of Claim 4, wherein the control arrangement contains a rule or rule set associated with governance of at least one sub-stream or object.

9. The player of Claim 8, wherein the rule or rule set is delivered from an external source.

10. The player of Claim 9, wherein the rule or rule set is delivered as part of the digital bit stream.

11. The player of Claim 8, wherein the rule or rule set specifies conditions under which the governed sub-stream or object may be decrypted.

12. The player of Claim 8, wherein the rule or rule set governs at least one aspect of access to or use of the governed sub-stream or object.

13. The player of Claim 12, wherein the governed aspect includes making copies of the governed sub-stream or object.

14. The player of Claim 12, wherein the governed aspect includes transmitting the governed sub-stream or object through a digital output port.

15. The player of Claim 14, wherein the rule or rule set specifies that the governed sub-stream or object can be transferred to a second device, but rendering of the governed sub-stream or object must be disabled in the first device prior to or during the transfer.

16. The player of Claim 15, wherein the second device includes rendering capability, lacks at least one feature present in the streaming media player, and is at least somewhat more portable than the streaming media player.

17. The player of Claim 11, wherein the control arrangement contains at least two rules governing access to or use of the same governed sub-stream or object.

18. The player of Claim 17, wherein a first of the two rules was supplied by a first entity, and the second of the two rules was supplied by a second entity.

19. The player of Claim 18, wherein the first rule controls at least one aspect of operation of the second rule.

20. The player of Claim 12, wherein the governed aspect includes use of at least one budget.

21. The player of Claim 12, wherein the governed aspect includes a requirement that audit information be provided.

22. The player of Claim 1, wherein the control arrangement includes tamper resistance.

23. A digital bit stream including:

content information that is compressed and at least in part encrypted; and

a secure container including

governance information for the governance of at least one aspect of access to or use of at least a portion of the content information; and

a key for decryption of at least a portion of the encrypted content information.

24. The digital bit stream of Claim 23, wherein the content information is encoded in MPEG-4 format.

25. The digital bit stream of Claim 23, wherein the content information is encoded in MP3 format.

26. A method of rendering a protected digital bit stream including:

receiving the protected digital bit stream,

passing the protected digital bit stream to a media player,

the media player reading first header information identifying a plugin used to process the protected digital bit stream, the first header information indicating that a first plugin is required;

the media player calling the first plugin;

the media player passing the protected digital bit stream to the first plugin;

the first plugin decrypting at least a portion of the protected digital bit stream;

the first plugin reading second header information identifying a second plugin necessary in order to render the decrypted digital bit stream;

the first plugin calling the second plugin;

the first plugin passing the decrypted digital bit stream to the second plugin;

the second plugin processing the decrypted digital bit stream, the processing including decompressing at least a portion of the decrypted digital bit stream;

the second plugin passing the decrypted and processed digital bit stream to the media player; and

the media player enabling rendering of the decrypted and processed digital bit stream,

whereby the first plugin may be used in an architecture not designed for multiple stages of plugin processing.

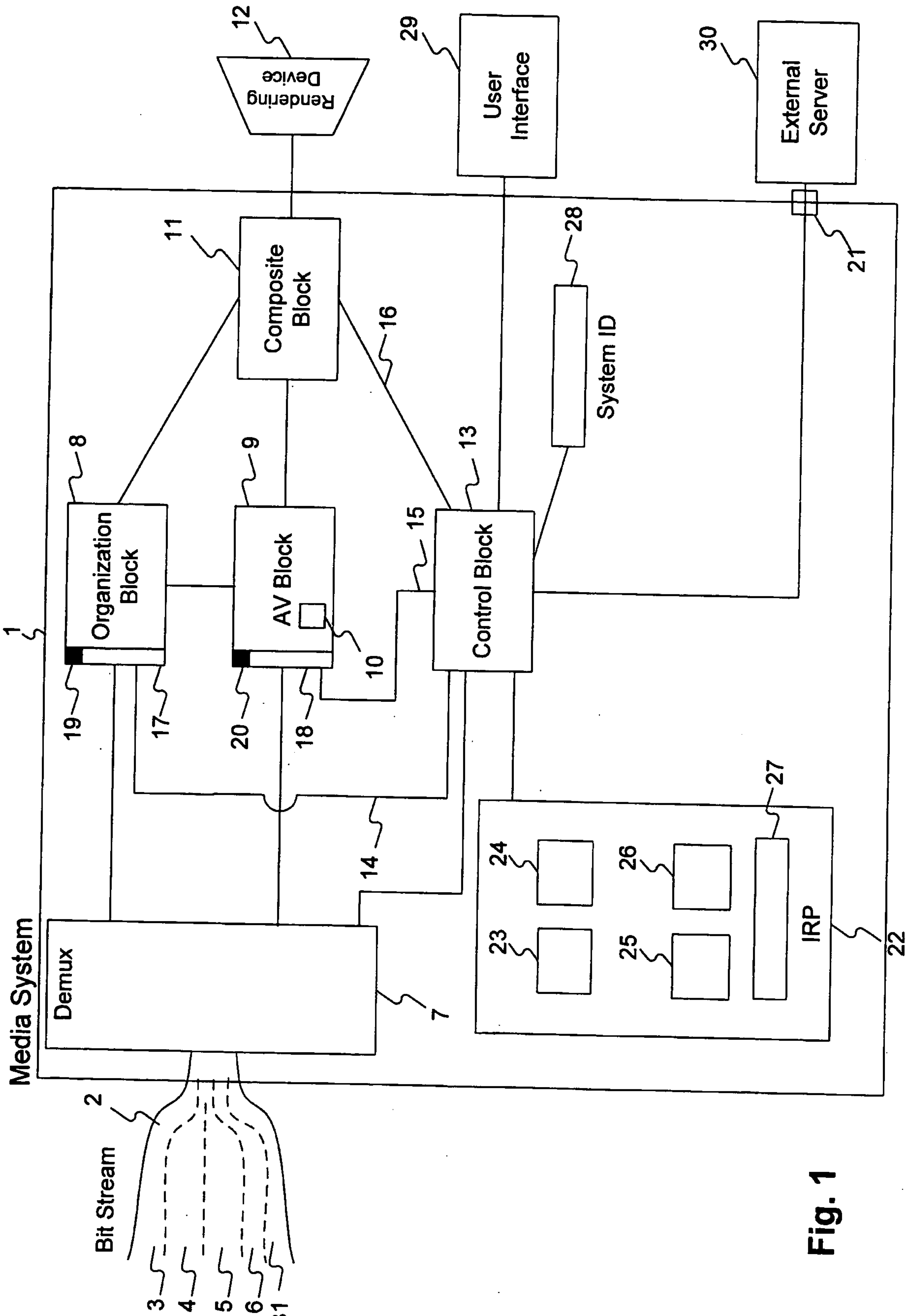

Media System
Demux
Organization Block
AV Block
Composite Block
Control Block
Rendering Device
User Interface
External Server
System ID
IRP
Bit Stream
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

Fig. 1

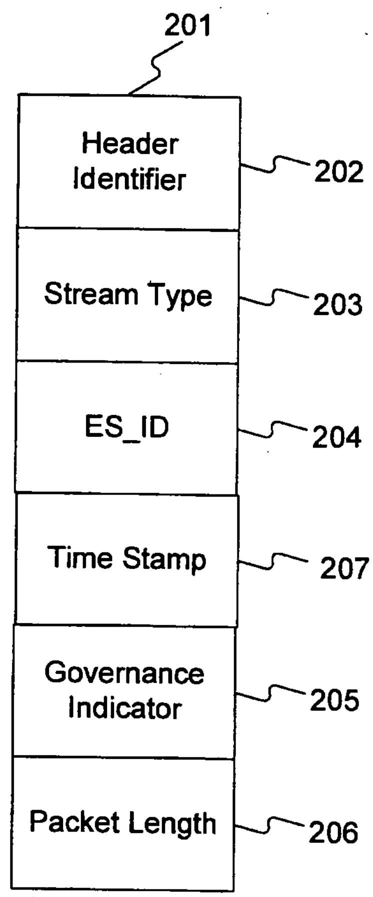

201
Header Identifier
202
Stream Type
203
ES_ID
204
Time Stamp
207
Governance Indicator
205
Packet Length
206

Fig. 2

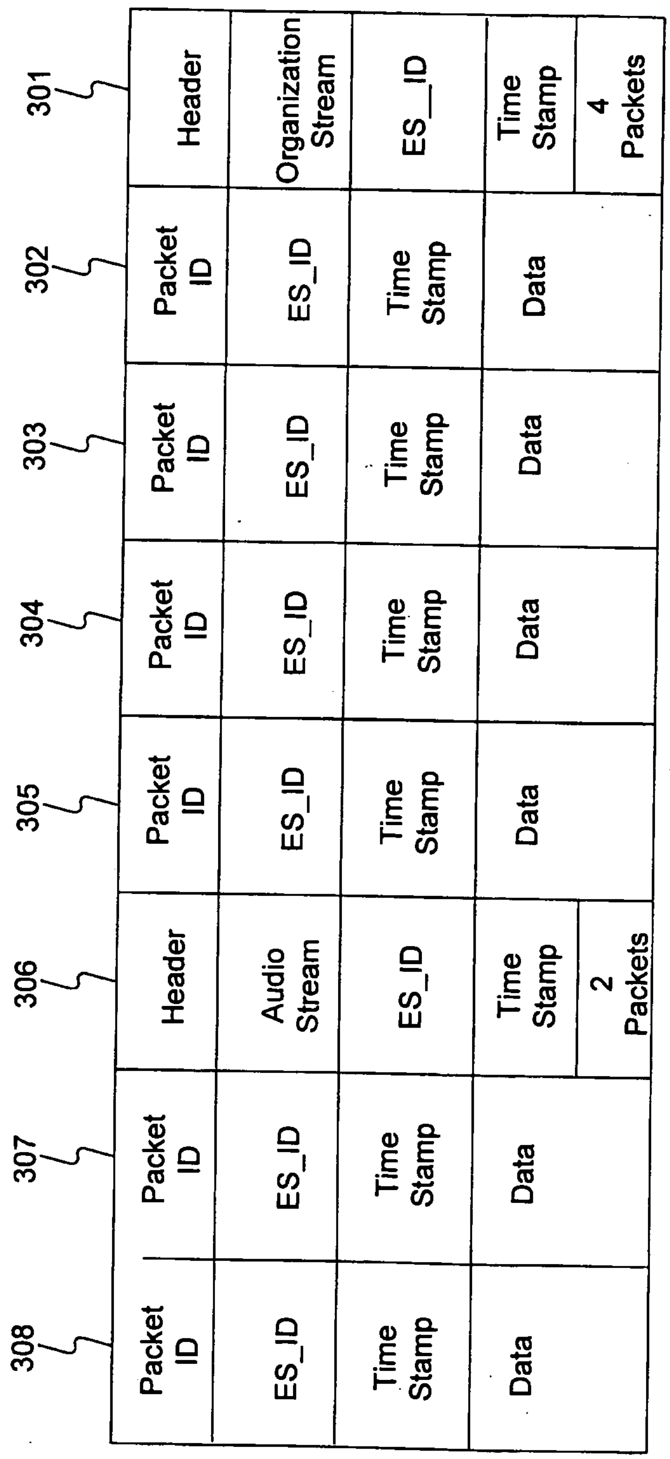

301
Header
Organization Stream
ES__ID
Time Stamp
4 Packets
302
Packet ID
ES_ID
Time Stamp
Data
303
Packet ID
ES_ID
Time Stamp
Data
304
Packet ID
ES_ID
Time Stamp
Data
305
Packet ID
ES_ID
Time Stamp
Data
306
Header
Audio Stream
ES_ID
Time Stamp
2 Packets
307
Packet ID
ES_ID
Time Stamp
Data
308
Packet ID
ES_ID
Time Stamp
Data

Fig. 3

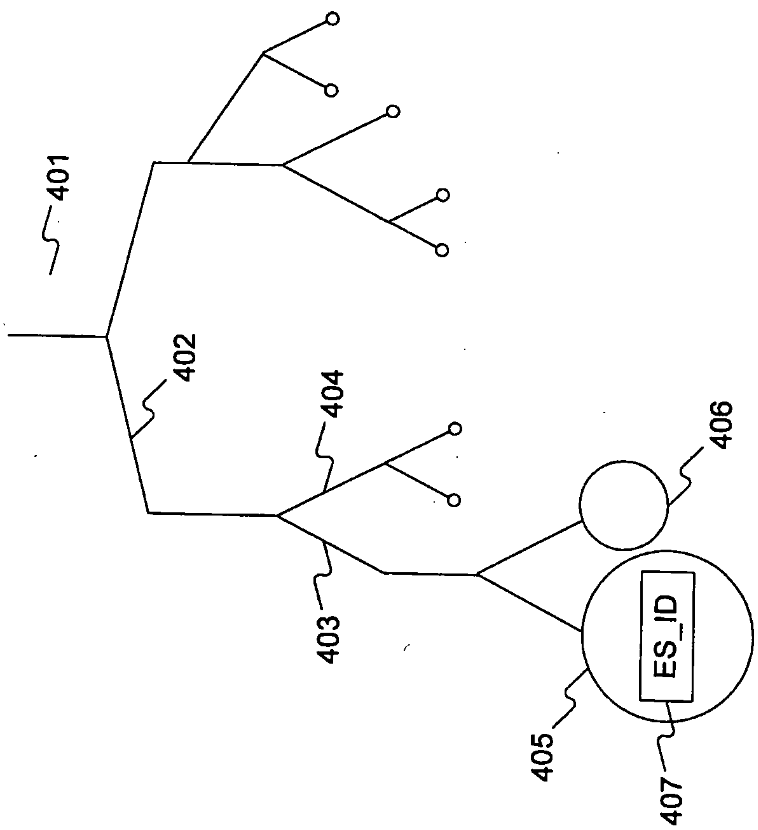


Fig. 4

Control Message

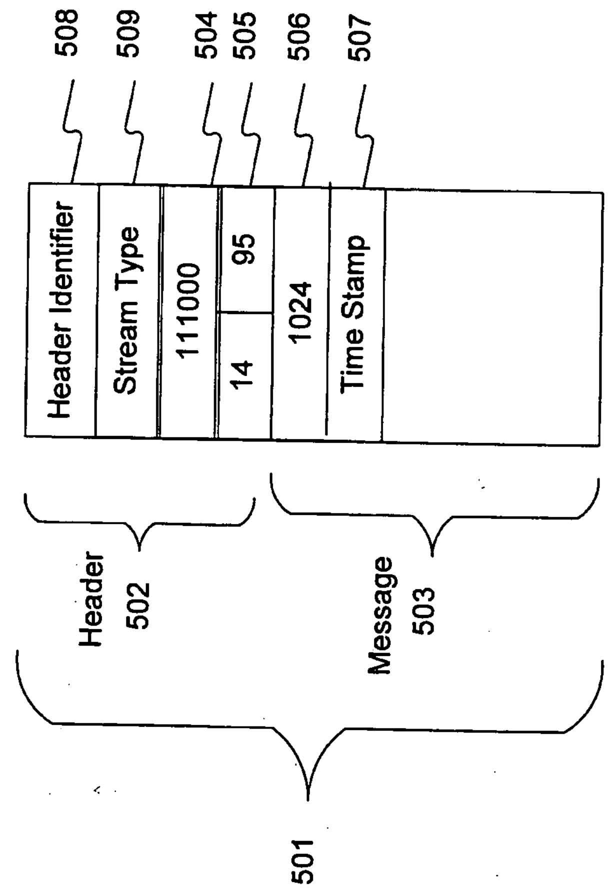


Fig. 5

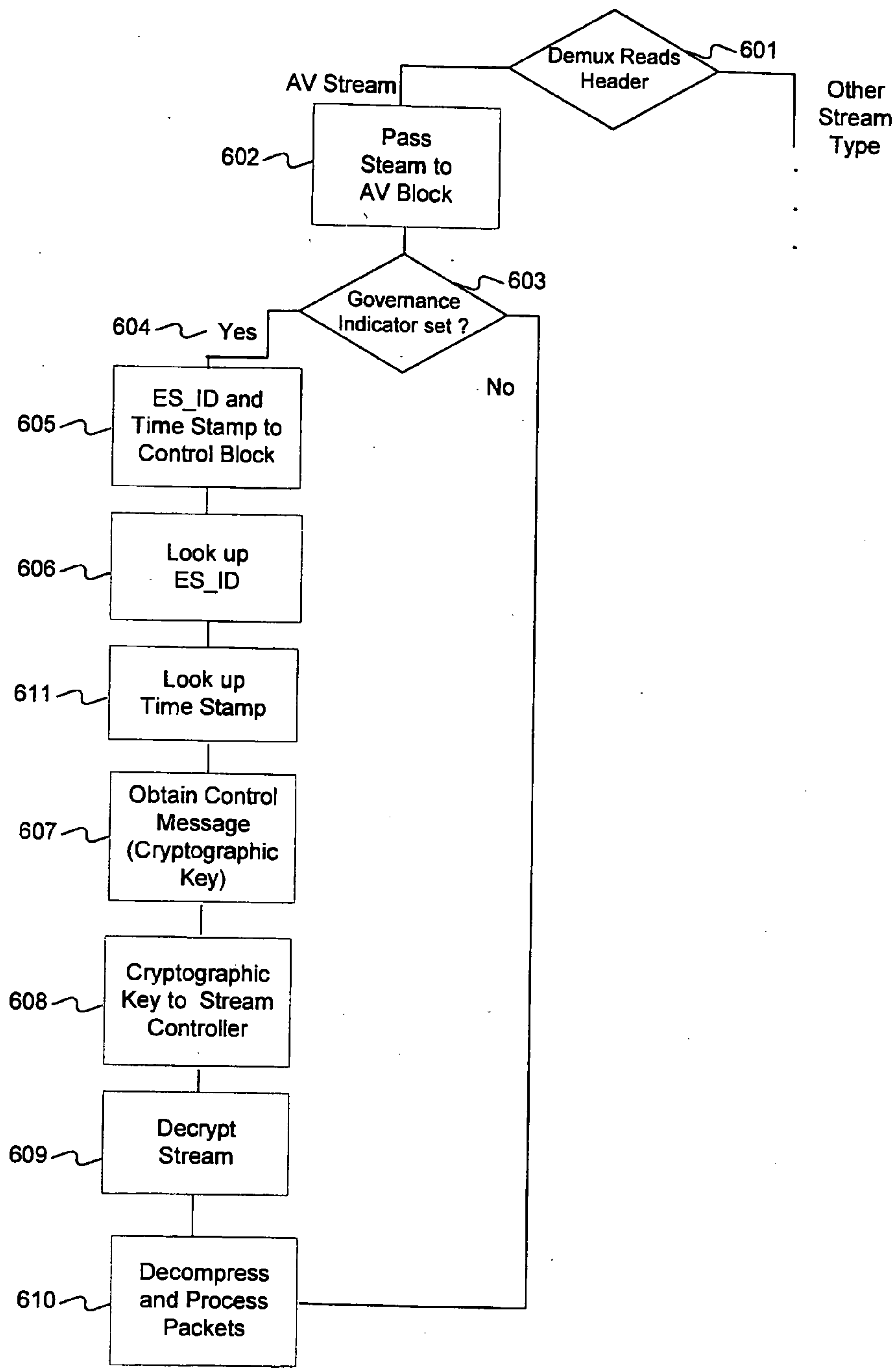

Demux Reads Header
601
AV Stream
Other Stream Type
602
Pass Steam to AV Block
603
Governance Indicator set ?
604
Yes
No
605
ES_ID and Time Stamp to Control Block
606
Look up ES_ID
611
Look up Time Stamp
607
Obtain Control Message (Cryptographic Key)
608
Cryptographic Key to Stream Controller
609
Decrypt Stream
610
Decompress and Process Packets

Fig. 6

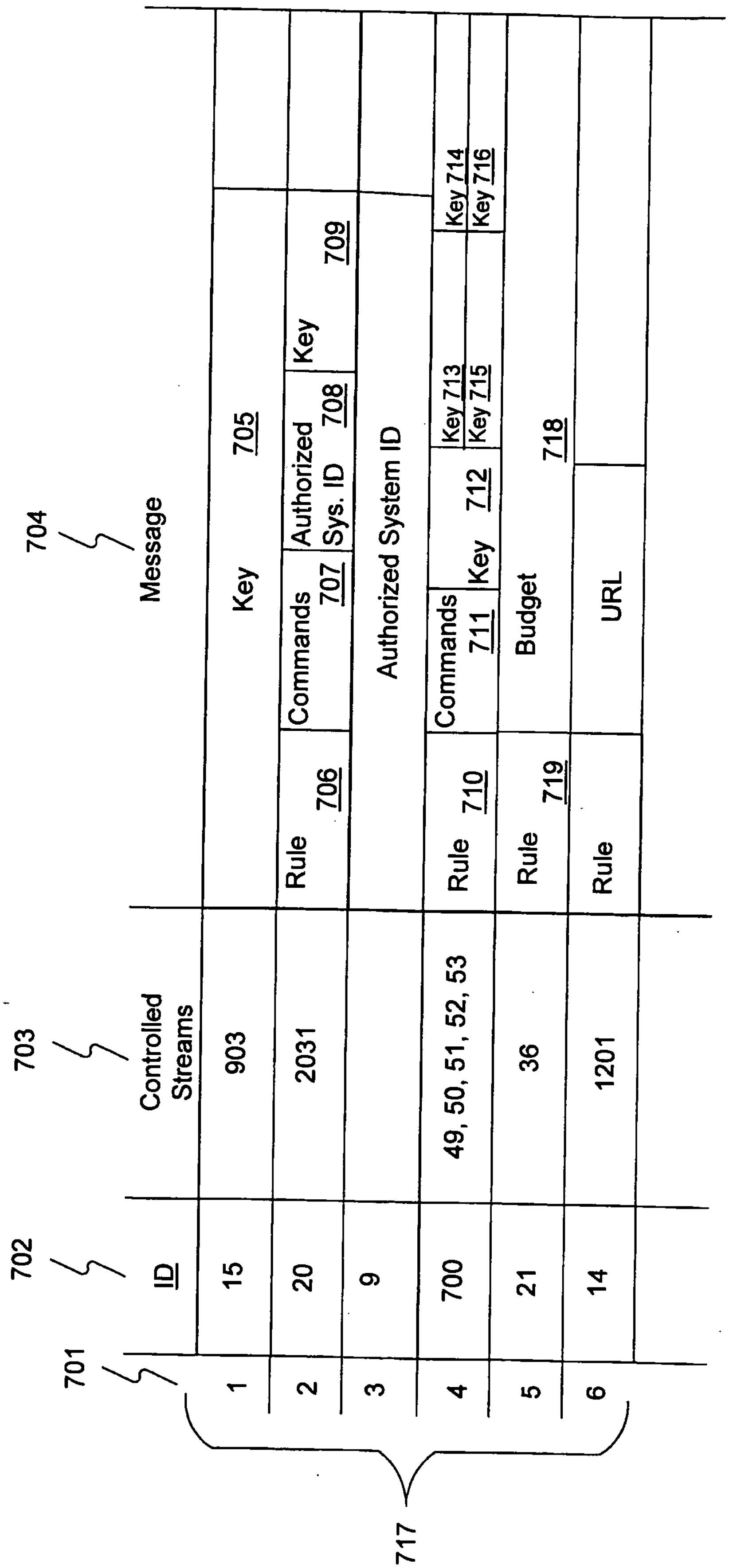

701
702
703
704
ID
Controlled Streams
Message
1
15
903
Key 705
2
20
2031
Rule 706
Commands 707
Authorized Sys. ID 708
Key 709
3
9
Authorized System ID
4
700
49, 50, 51, 52, 53
Rule 710
Commands 711
Key 712
Key 713
Key 714
Key 715
Key 716
5
21
36
Rule 719
Budget 718
6
14
1201
Rule
URL
717

Fig. 7

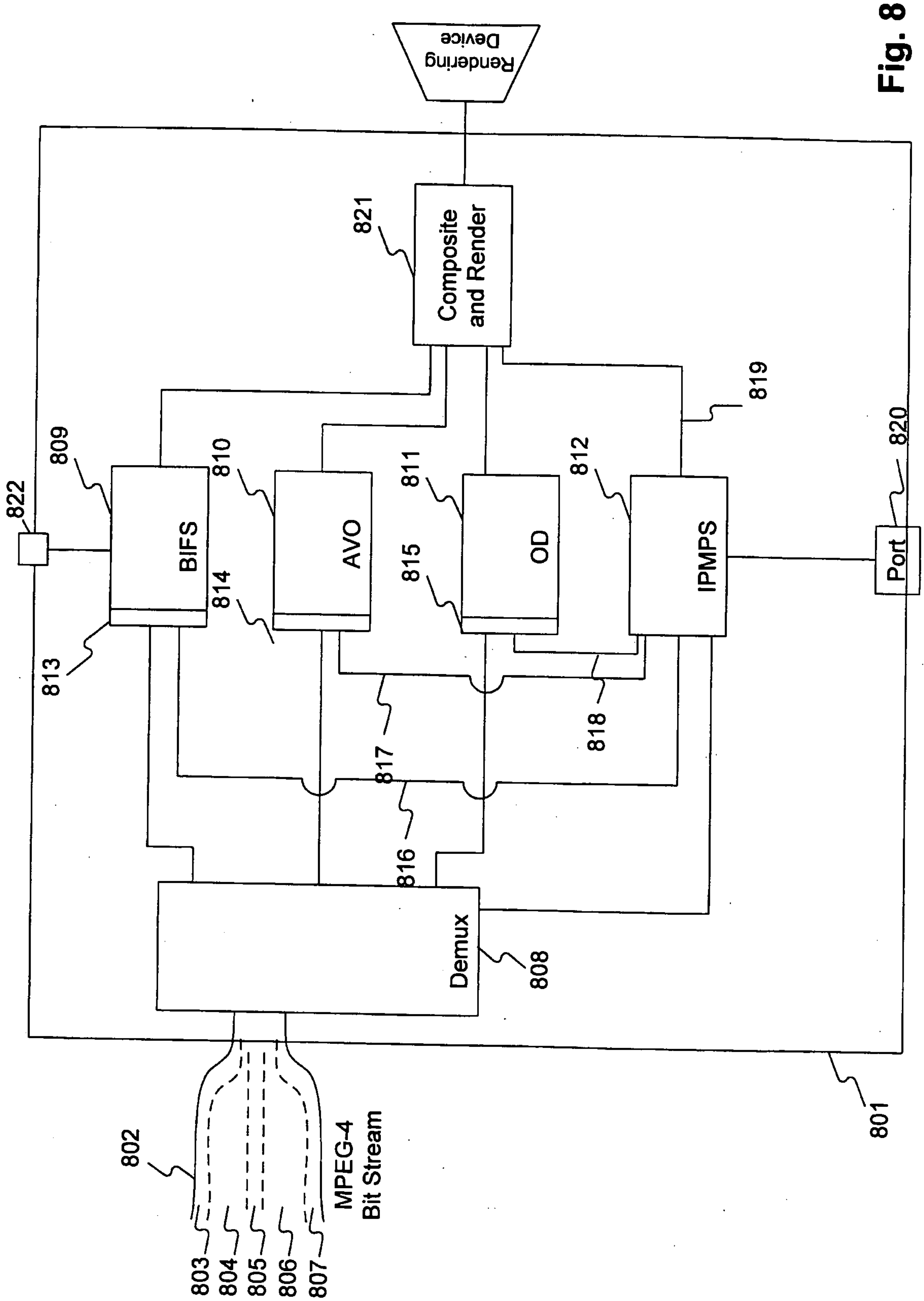

Rendering Device
Composite and Render
821
819
820
Port
822
809
BIFS
810
AVO
811
OD
812
IPMPS
813
814
815
817
818
816
Demux
808
801
802
MPEG-4 Bit Stream
803
804
805
806
807

Fig. 8

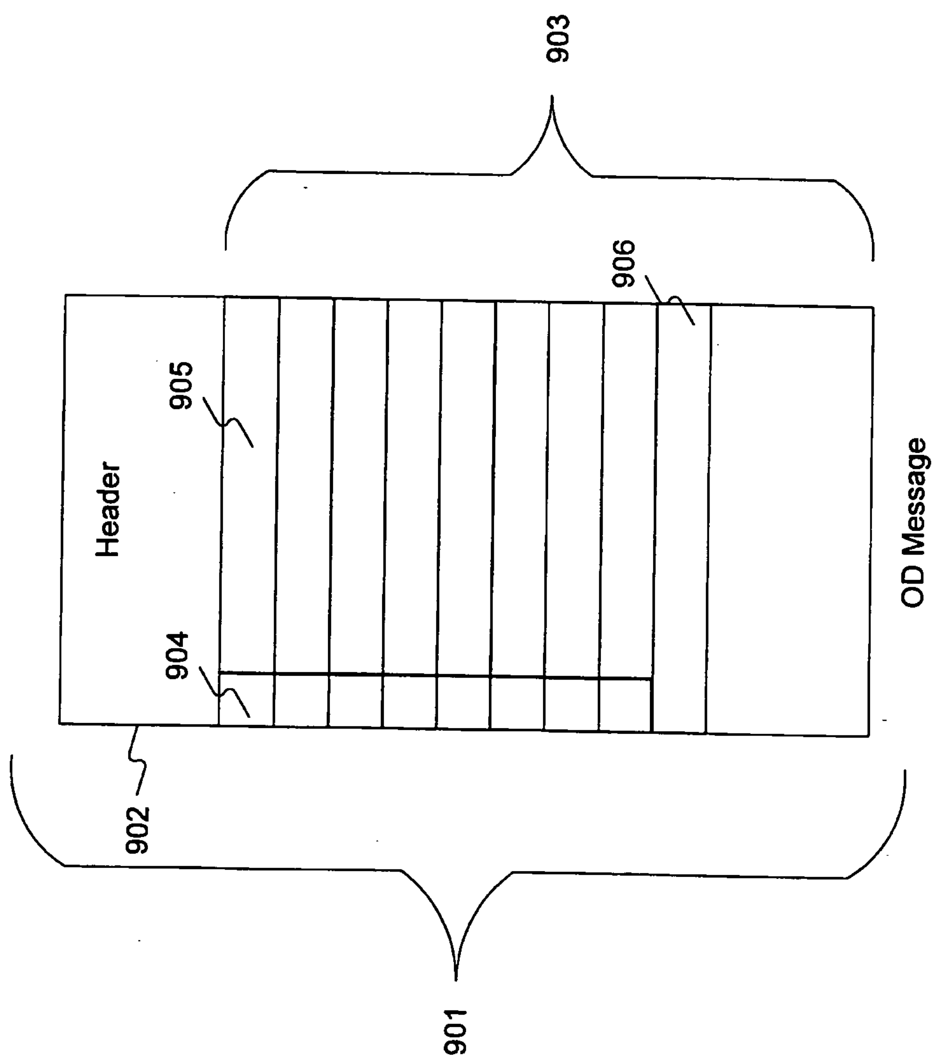


Fig. 9

IPMP Table

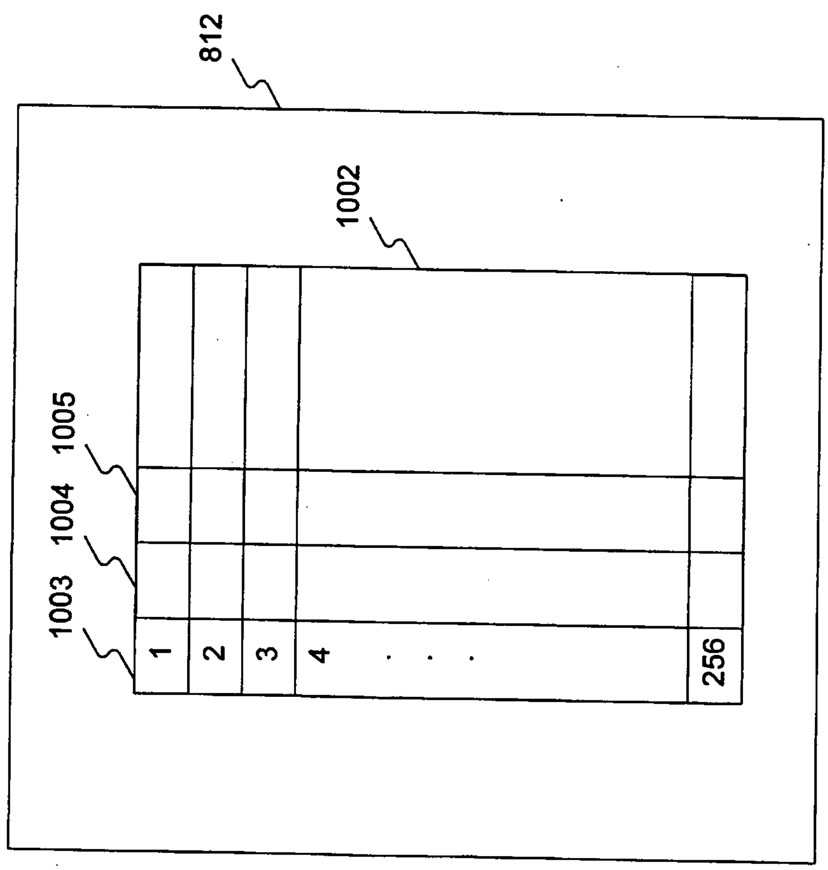


Fig. 10

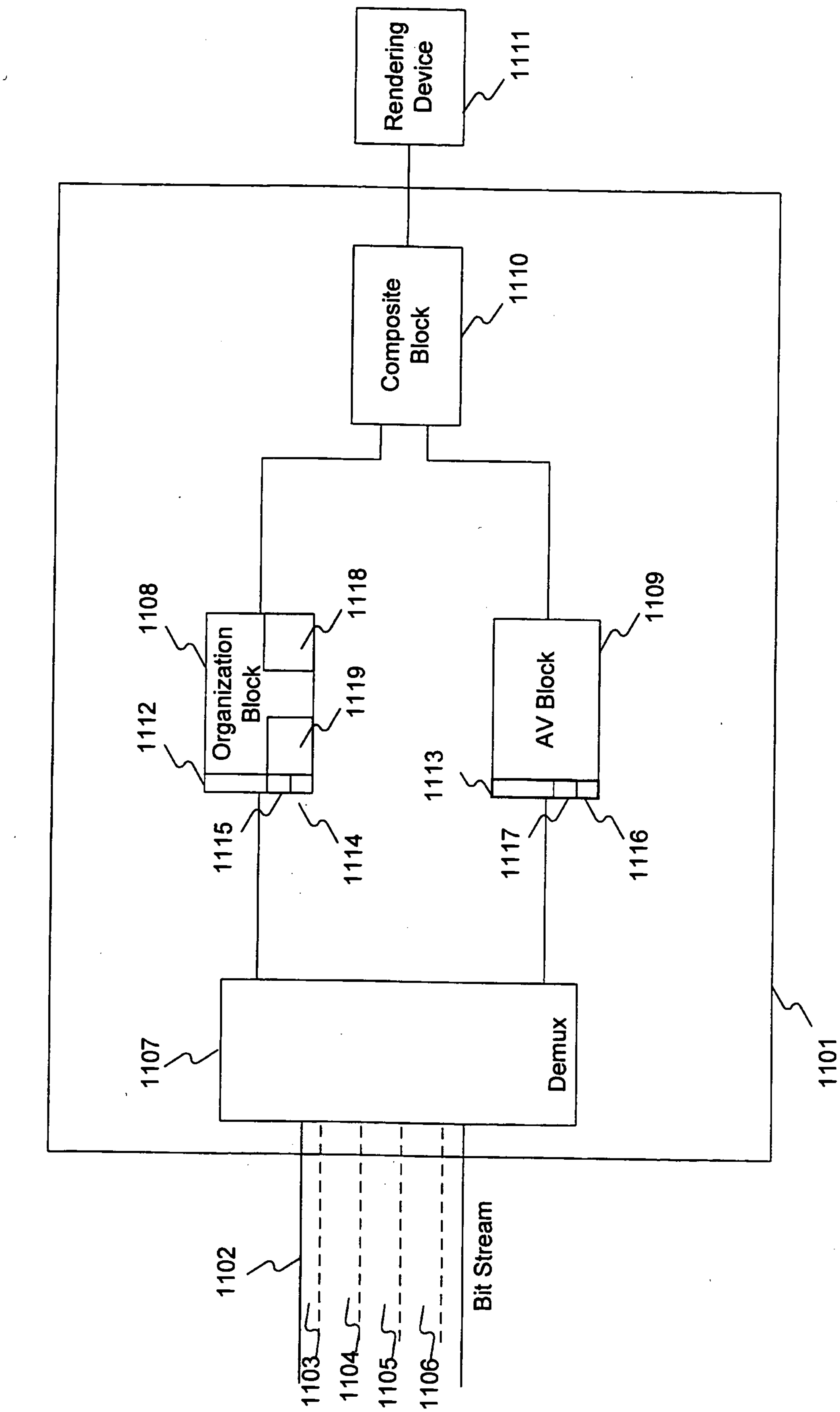

Rendering Device
1111
Composite Block
1110
Organization Block
1108
1118
1119
1112
1115
1114
AV Block
1109
1113
1117
1116
Demux
1107
1101
1102
Bit Stream
1103
1104
1105
1106

Fig. 11

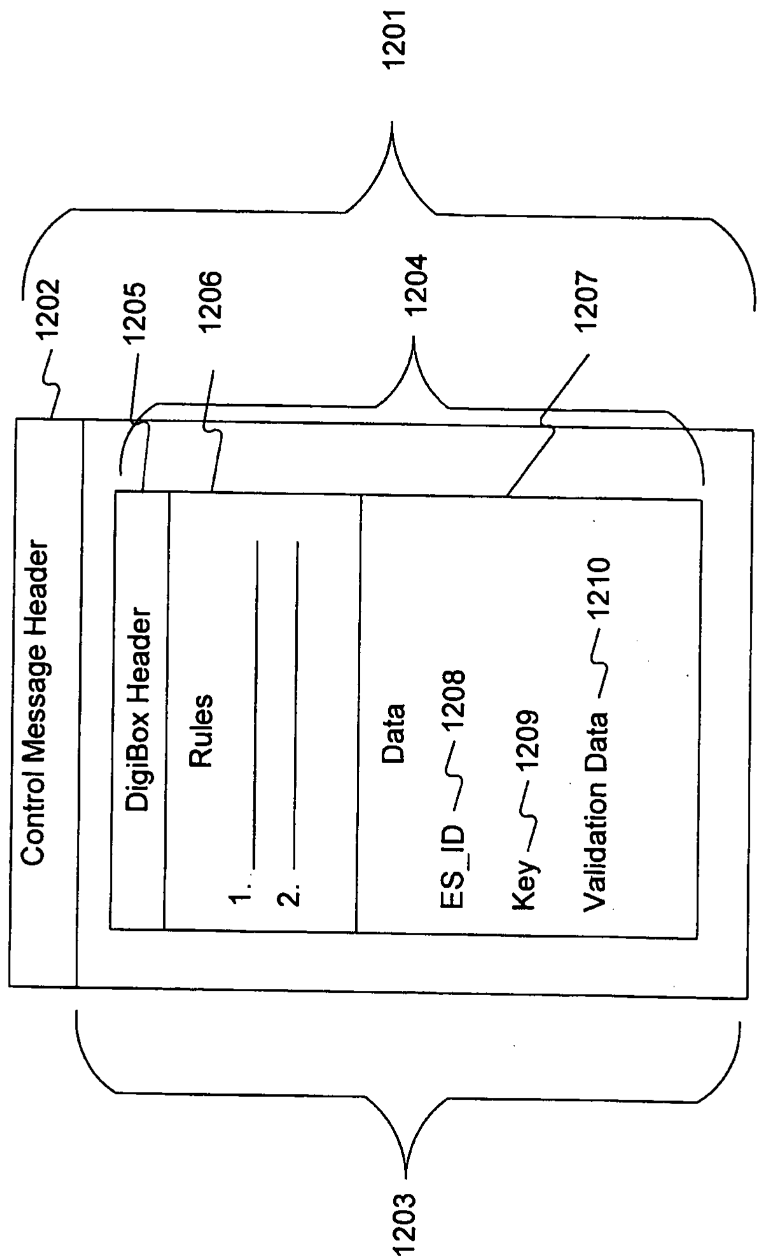

1201
1202
1205
1206
1204
1207
Control Message Header
DigiBox Header
Rules
1.
2.
Data
ES_ID
1208
Key
1209
Validation Data
1210
1203

Fig. 12

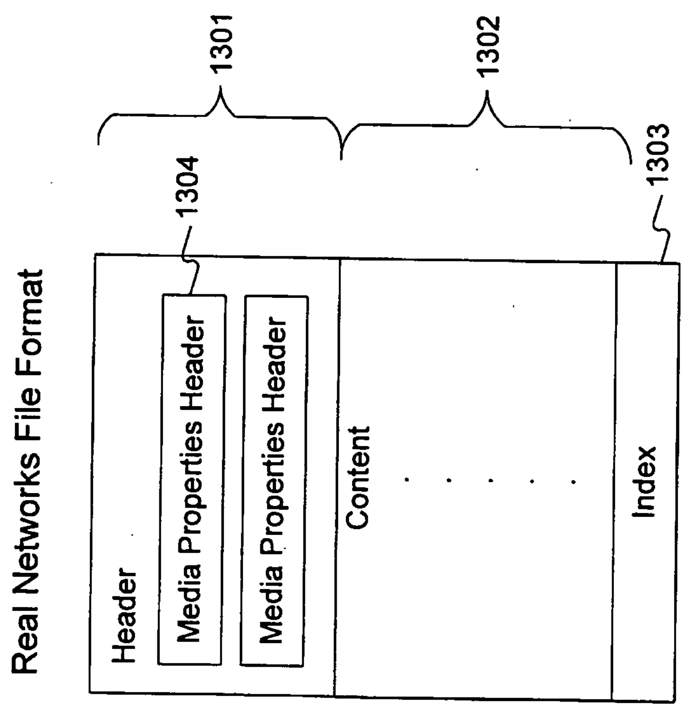

Real Networks File Format
Header
Media Properties Header
Media Properties Header
Content
Index
1301
1302
1303
1304

Fig. 13

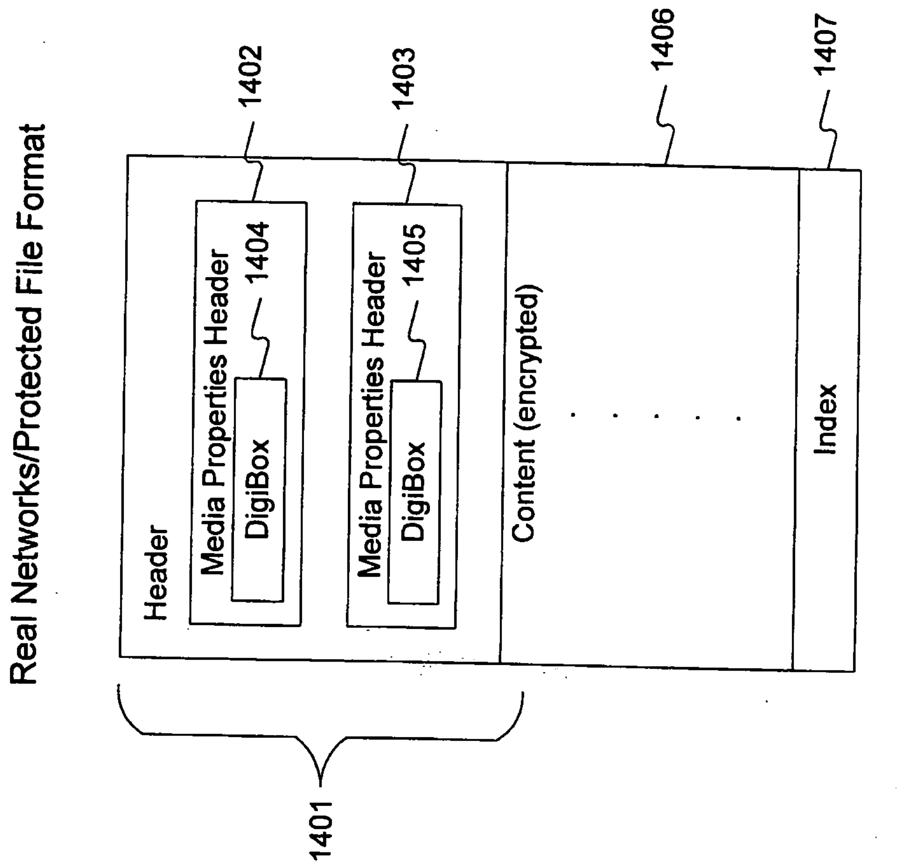

Real Networks/Protected File Format
Header
Media Properties Header
DigiBox
1404
1402
Media Properties Header
DigiBox
1405
1403
Content (encrypted)
1406
Index
1407
1401

Fig. 14

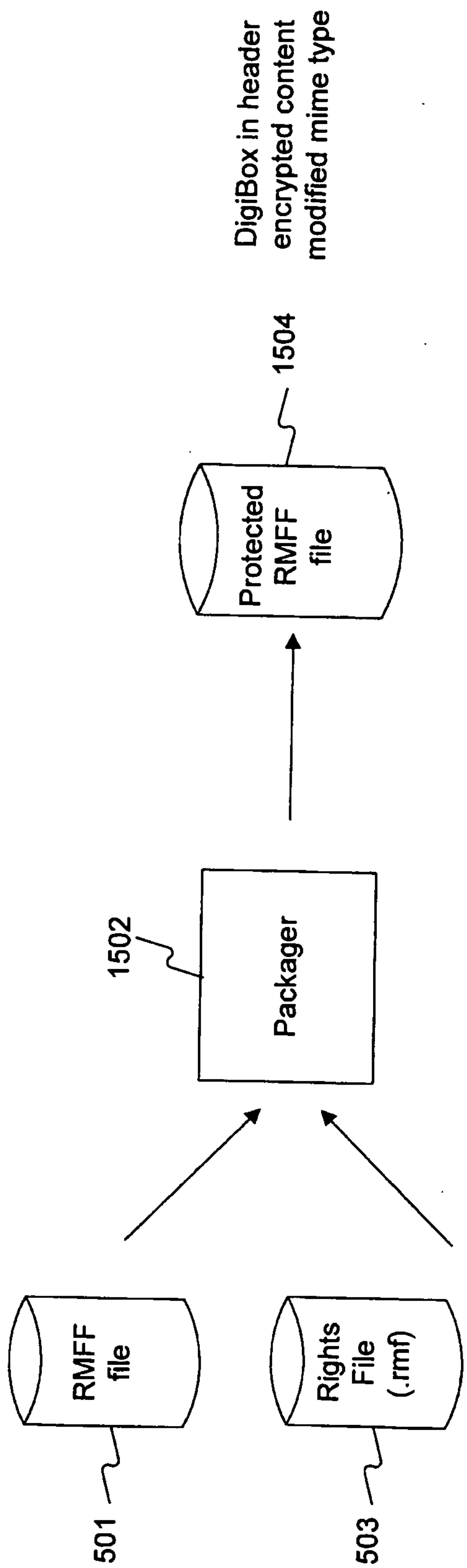

RMFF file
1501
Rights File (.rmf)
1503
Packager
1502
Protected RMFF file
1504
DigiBox in header
encrypted content
modified mime type

Fig. 15

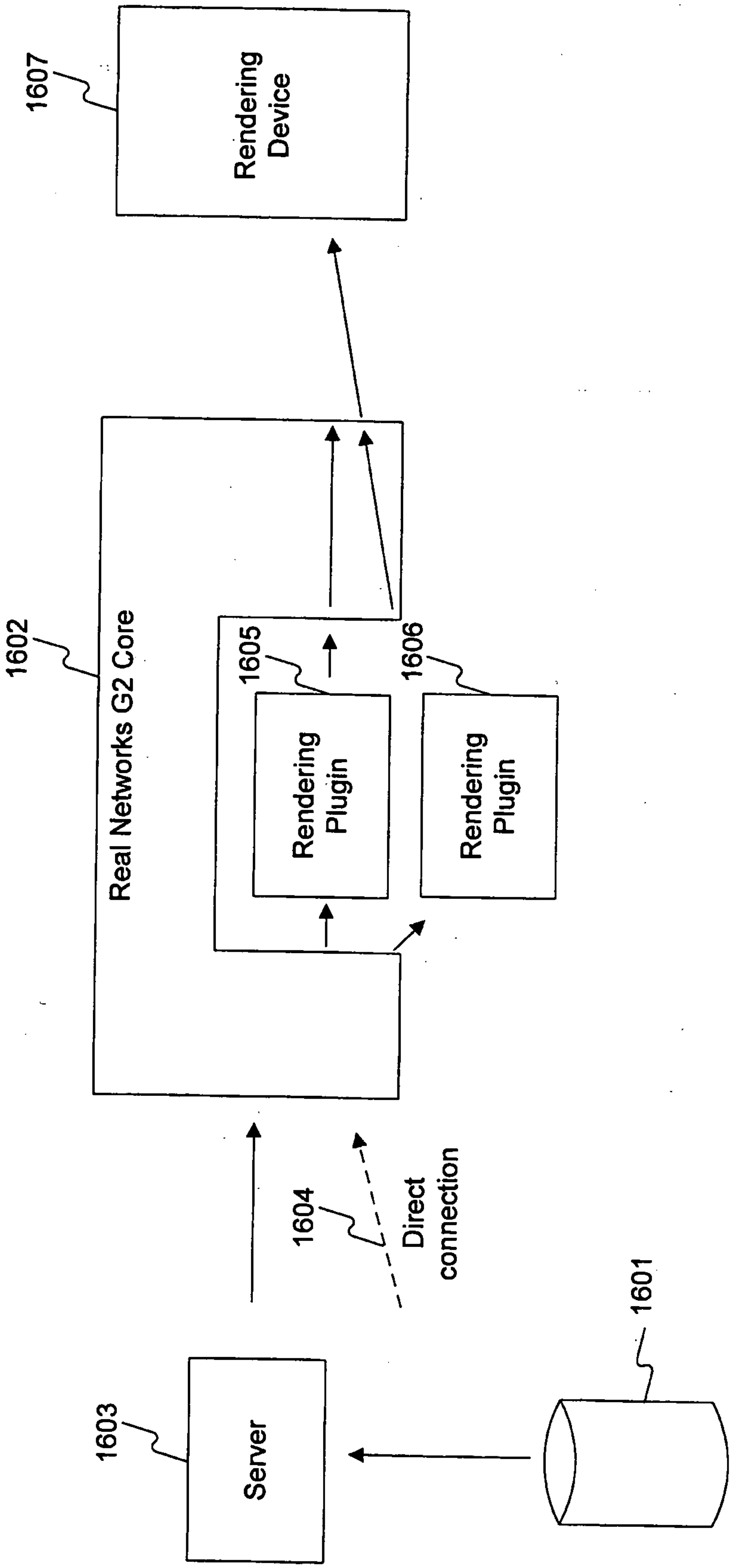

1607
Rendering Device
1602
Real Networks G2 Core
1605
Rendering Plugin
1606
Rendering Plugin
1604
Direct connection
1603
Server
1601

Fig. 16

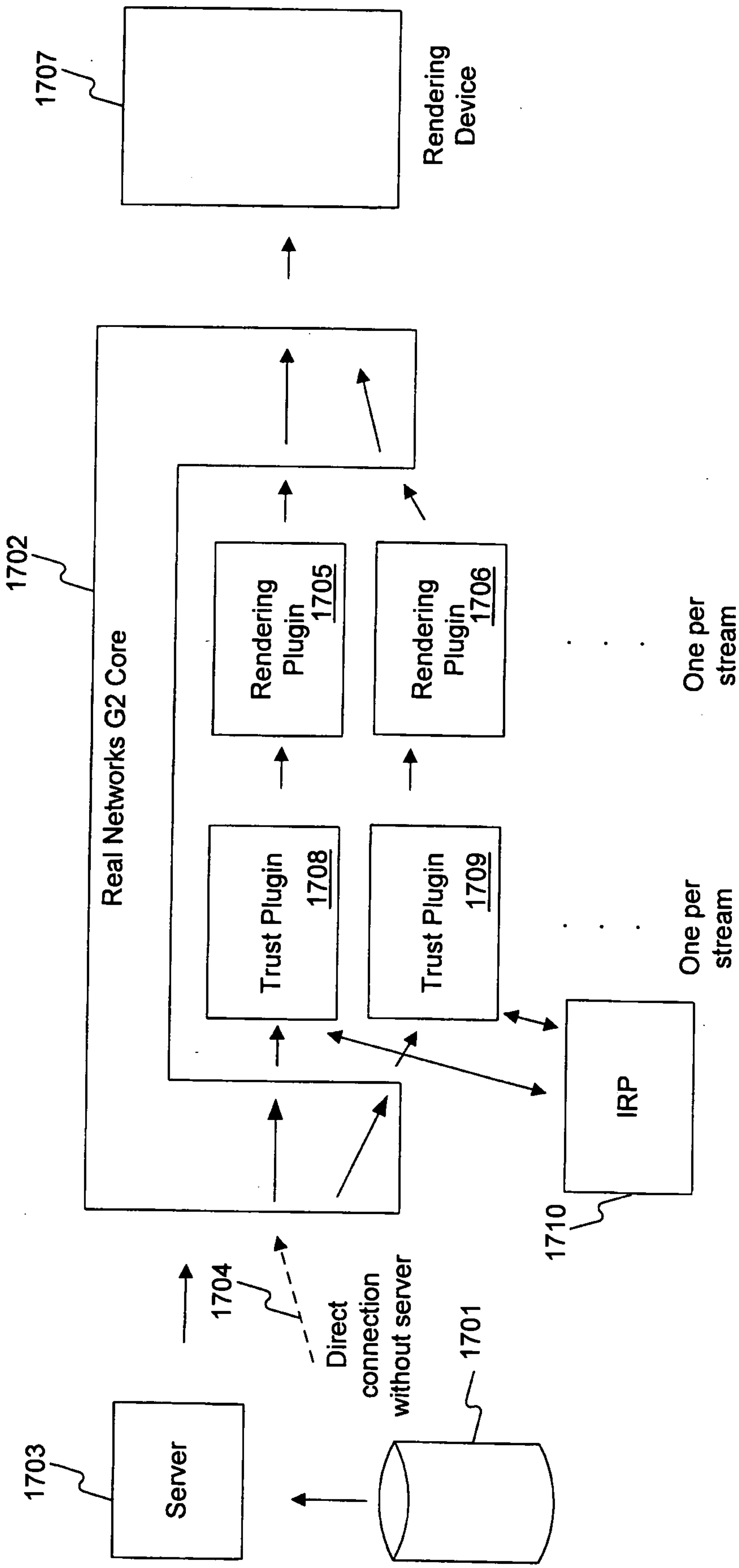

1707
Rendering Device
1702
Real Networks G2 Core
Rendering Plugin 1705
Rendering Plugin 1706
One per stream
Trust Plugin 1708
Trust Plugin 1709
One per stream
IRP
1710
1704
Direct connection without server
1701
1703
Server

Fig. 17

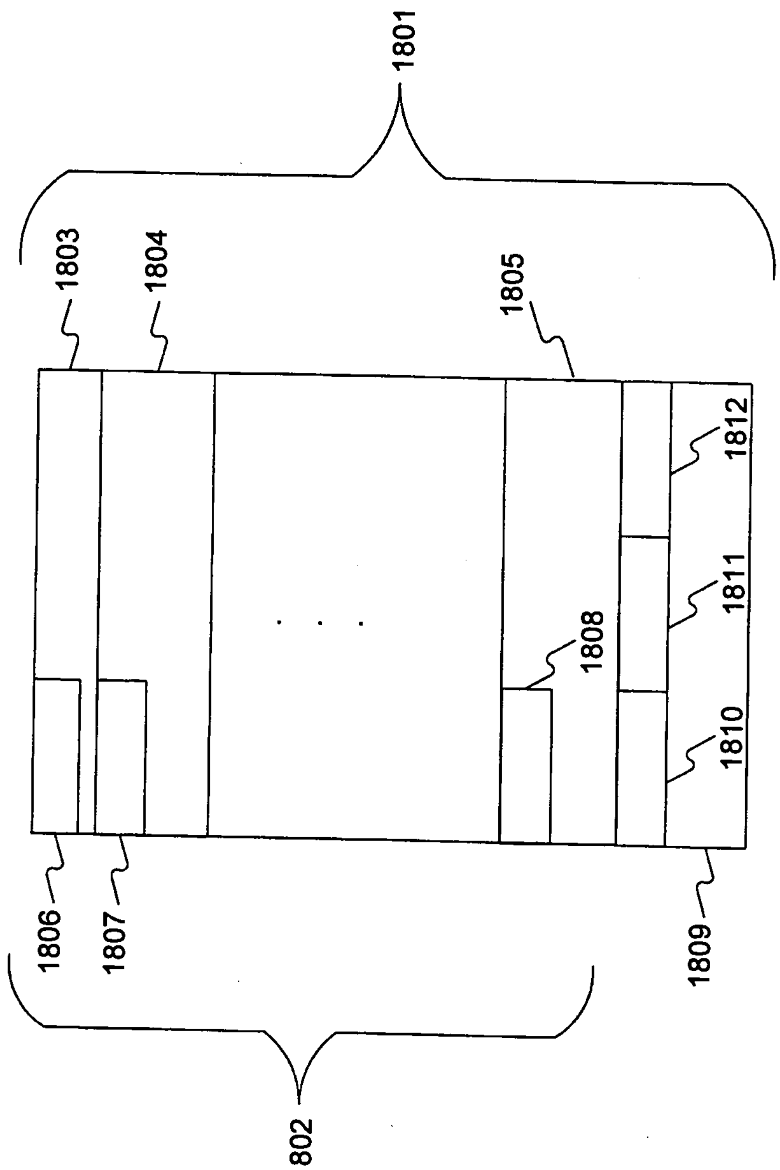
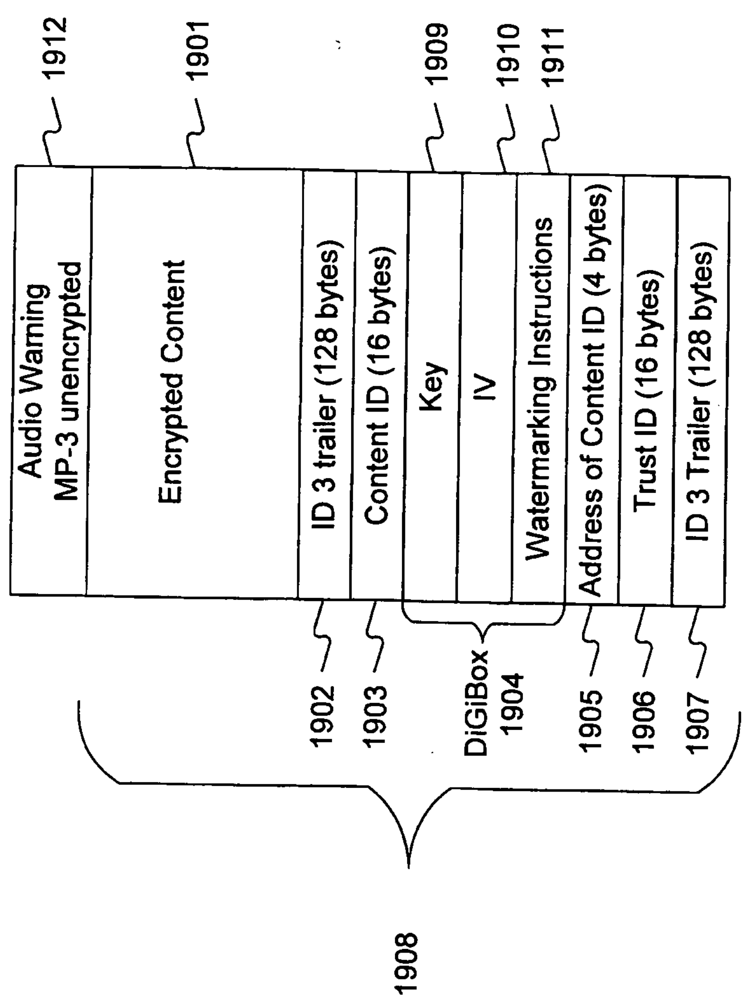


Fig. 18



Protected MP3 Format

Fig. 19

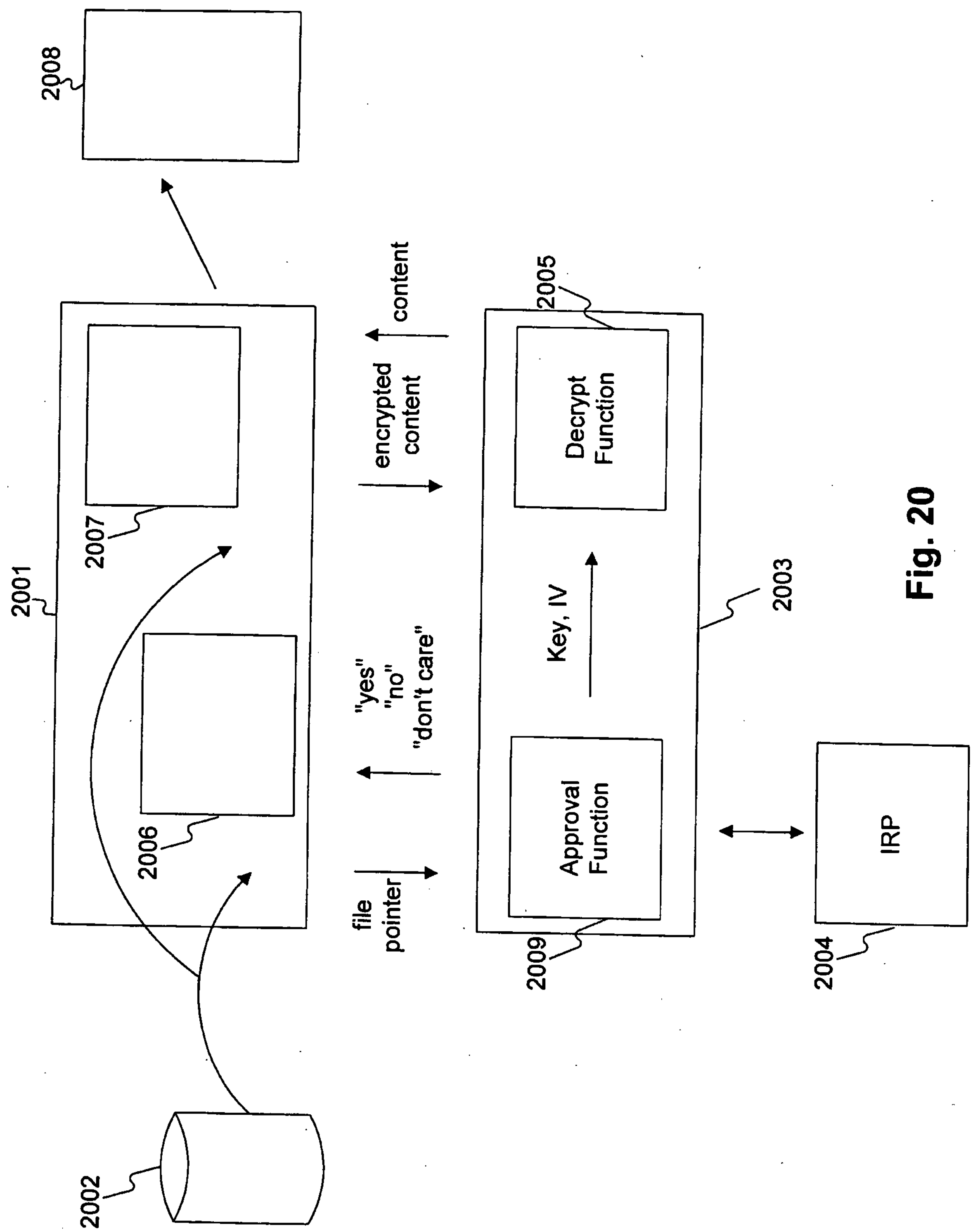

2008
2001
2007
2006
2002
content
encrypted content
"yes"
"no"
"don't care"
file pointer
2005
Decrypt Function
Key, IV
Approval Function
2003
2009
IRP
2004

Fig. 20

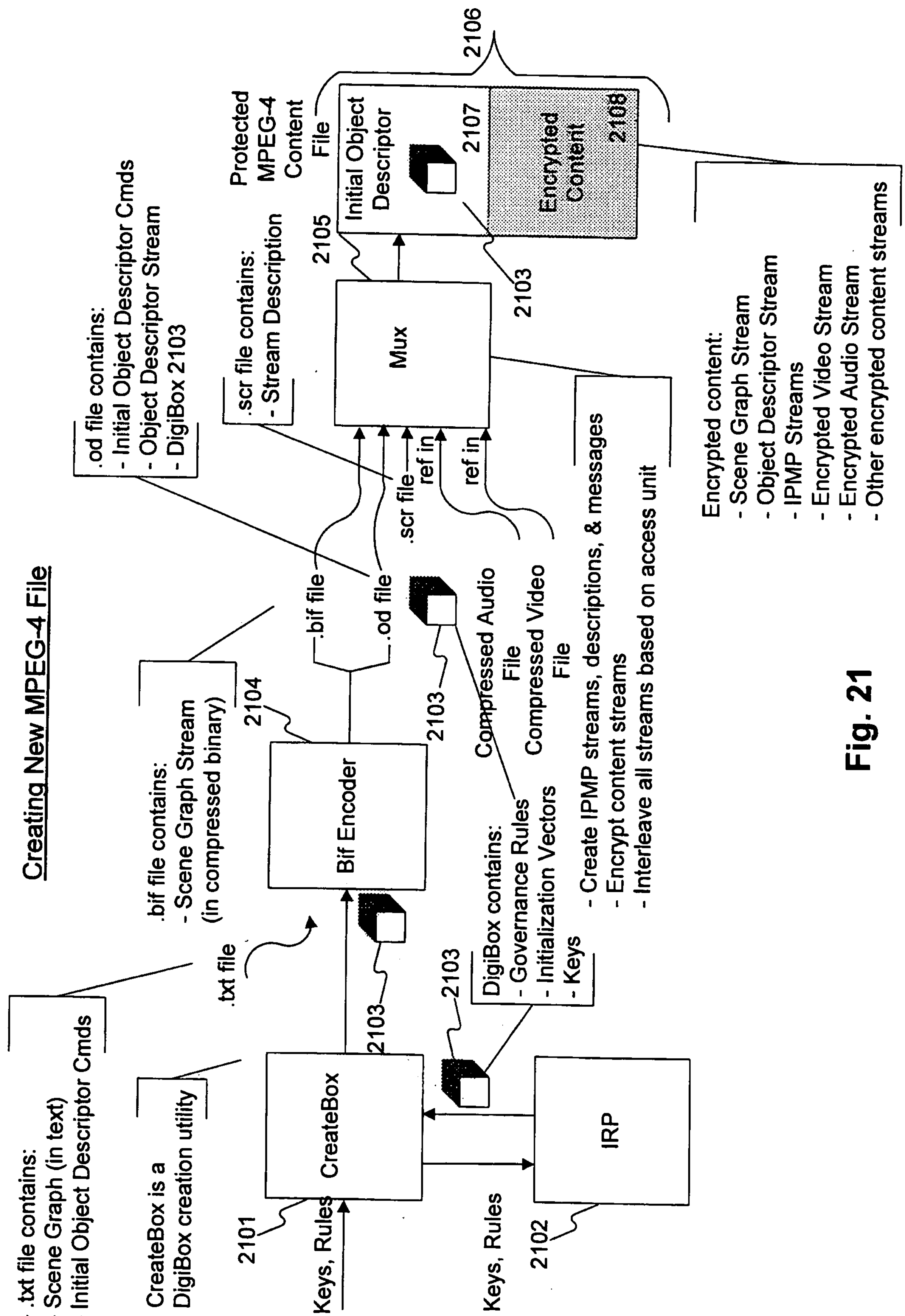

Creating New MPEG-4 File
- .txt file contains:
- Scene Graph (in text)
- Initial Object Descriptor Cmds
CreateBox is a
DigiBox creation utility
Keys, Rules
2101
CreateBox
Keys, Rules
2102
IRP
2103
.txt file
DigiBox contains:
- Governance Rules
- Initialization Vectors
- Keys
.bif file contains:
- Scene Graph Stream
(in compressed binary)
Bif Encoder
2104
.bif file
.od file
Compressed Audio
File
Compressed Video
File
- Create IPMP streams, descriptions, & messages
- Encrypt content streams
- Interleave all streams based on access unit
.od file contains:
- Initial Object Descriptor Cmds
- Object Descriptor Stream
- DigiBox 2103
.scr file contains:
- Stream Description
.scr file
ref in
ref in
Mux
2105
Protected
MPEG-4
Content
File
Initial Object
Descriptor
2107
Encrypted
Content
2108
2106
Encrypted content:
- Scene Graph Stream
- Object Descriptor Stream
- IPMP Streams
- Encrypted Video Stream
- Encrypted Audio Stream
- Other encrypted content streams

Fig. 21

Repackaging Existing MPEG-4 Files

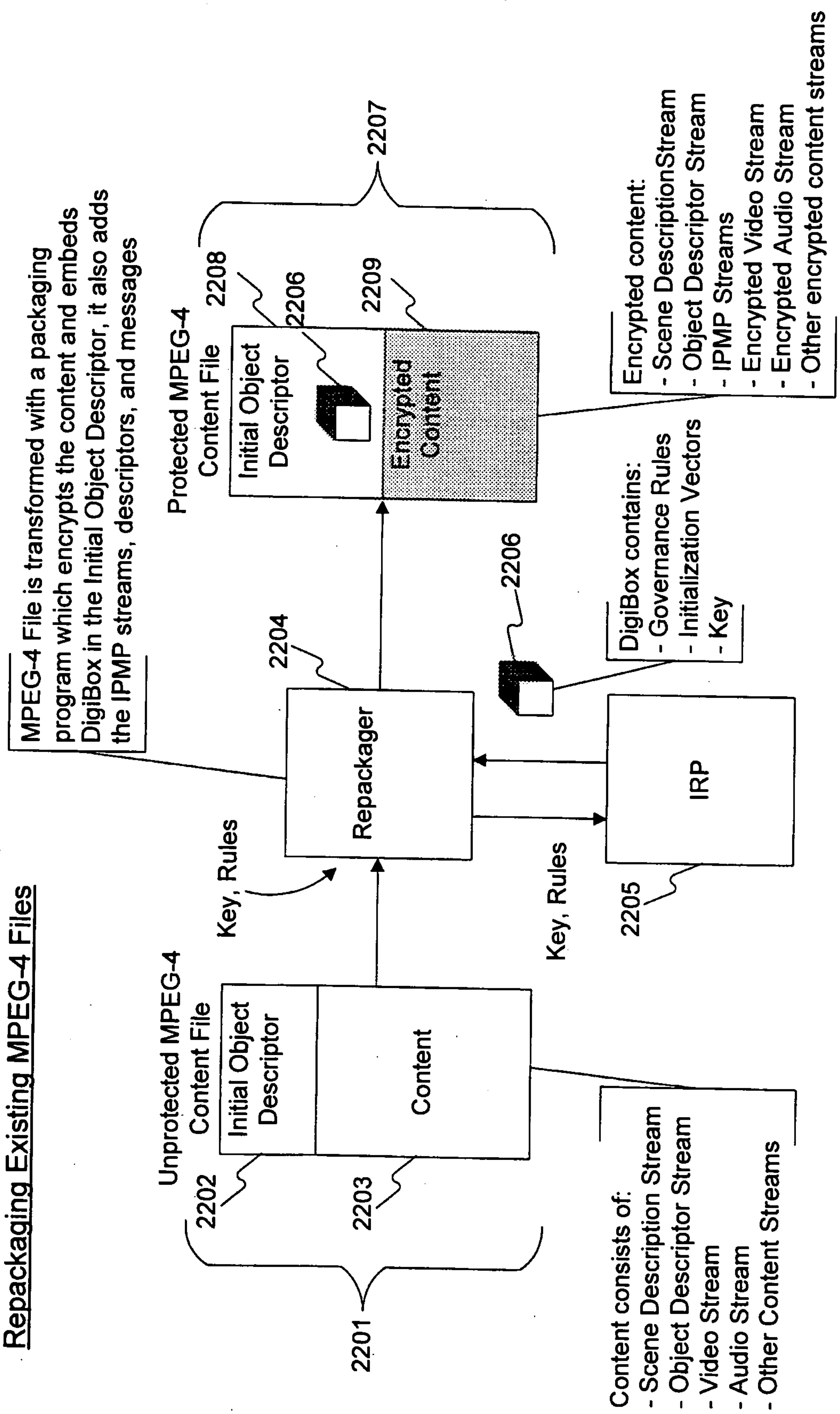


Fig. 22

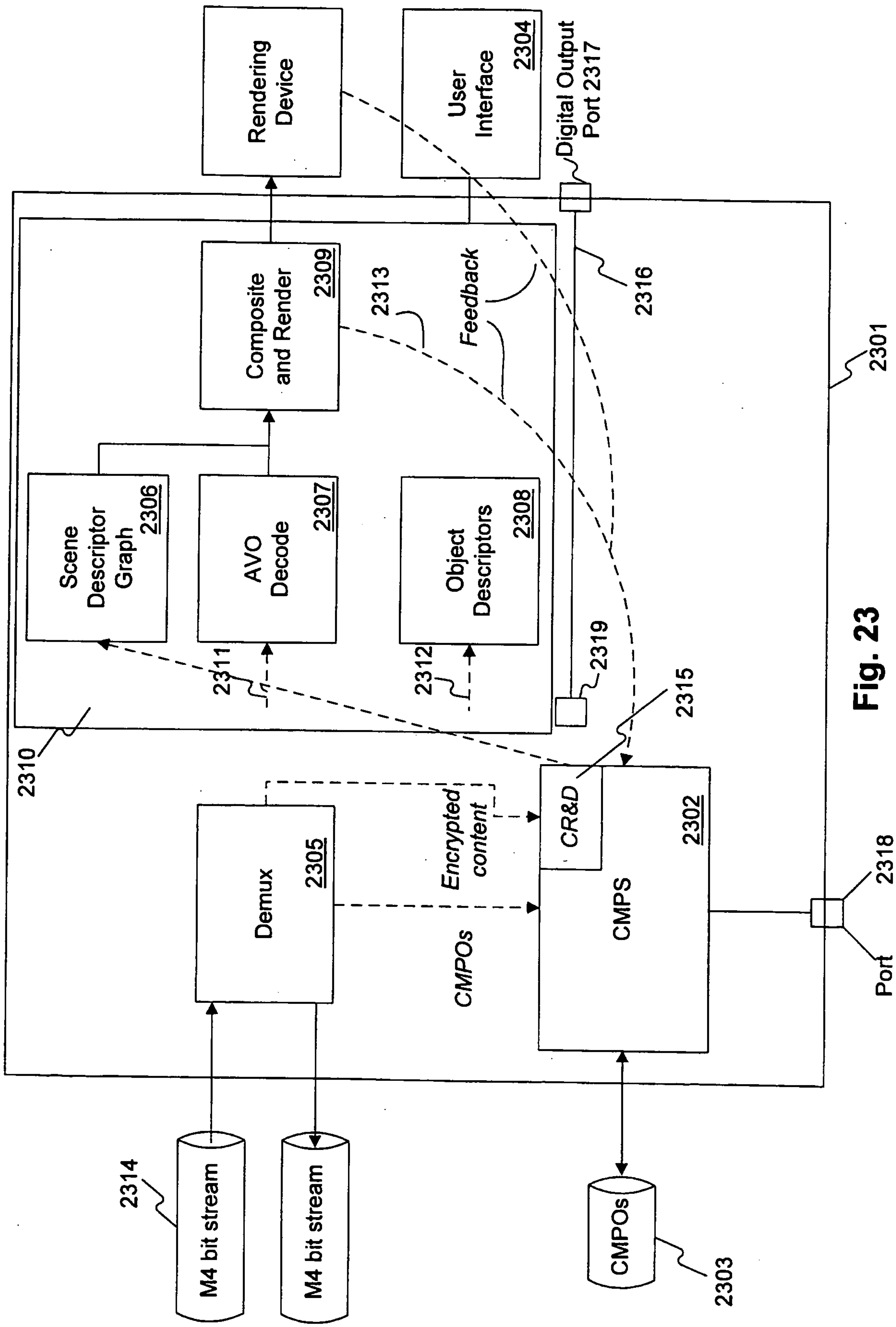

Rendering Device
User Interface 2304
Digital Output Port 2317
Composite and Render 2309
2313
Feedback
2316
2301
Scene Descriptor Graph 2306
AVO Decode 2307
Object Descriptors 2308
2311
2312
2319
2315
2310
CR&D
CMPS 2302
Demux 2305
Encrypted content
CMPOs
2318
Port
2314
M4 bit stream
M4 bit stream
CMPOs
2303

Fig. 23

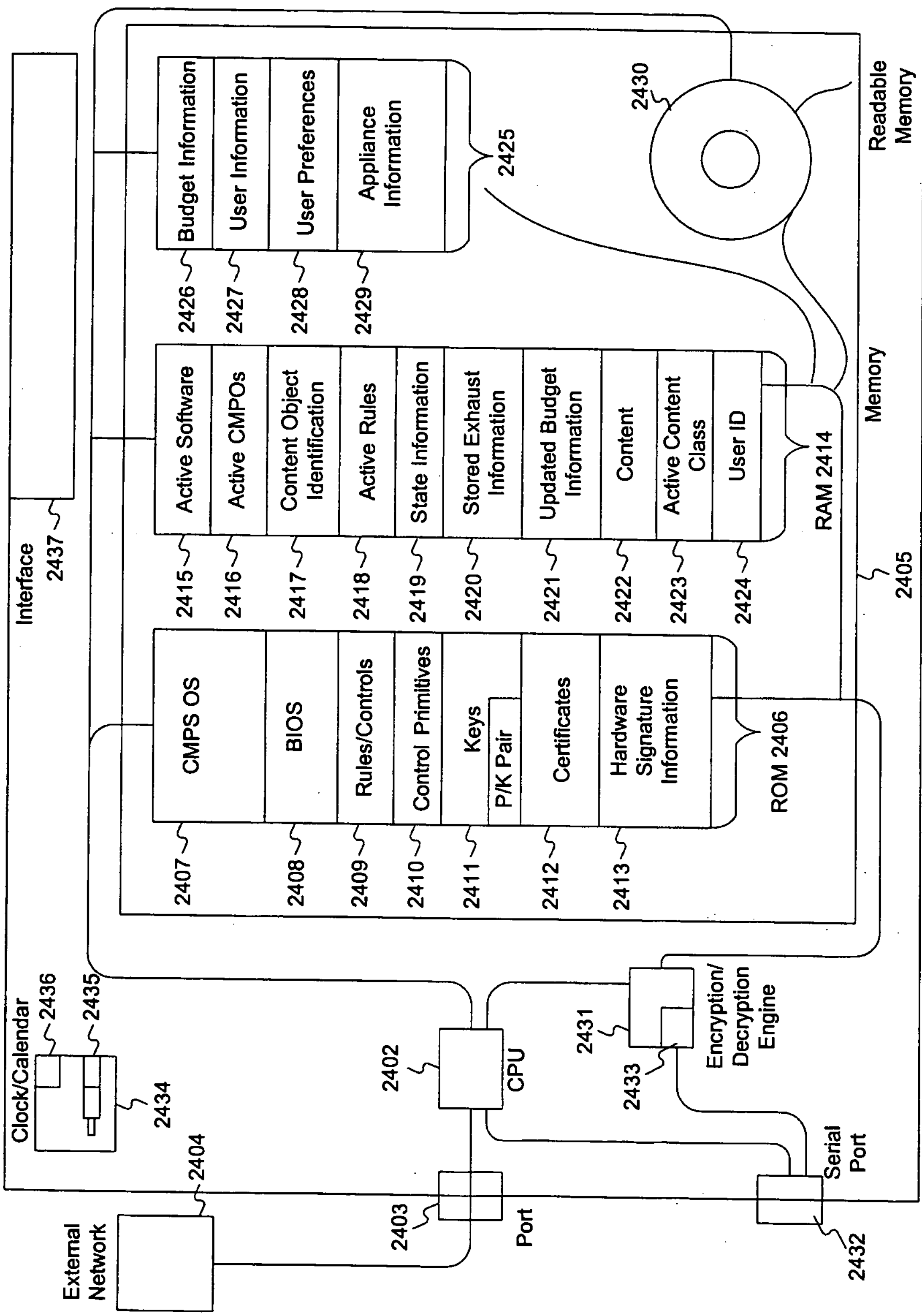

Interface
2437
Budget Information
User Information
User Preferences
Appliance Information
2425
2426
2427
2428
2429
2430
Readable Memory
Active Software
Active CMPOs
Content Object Identification
Active Rules
State Information
Stored Exhaust Information
Updated Budget Information
Content
Active Content Class
User ID
RAM 2414
Memory
2401
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2405
CMPS
CMPS OS
BIOS
Rules/Controls
Control Primitives
Keys
P/K Pair
Certificates
Hardware Signature Information
ROM 2406
2407
2408
2409
2410
2411
2412
2413
Clock/Calendar
2436
2435
2434
2404
External Network
2402
CPU
2403
Port
2431
2433
Encryption/ Decryption Engine
Serial Port
2432

Fig. 24

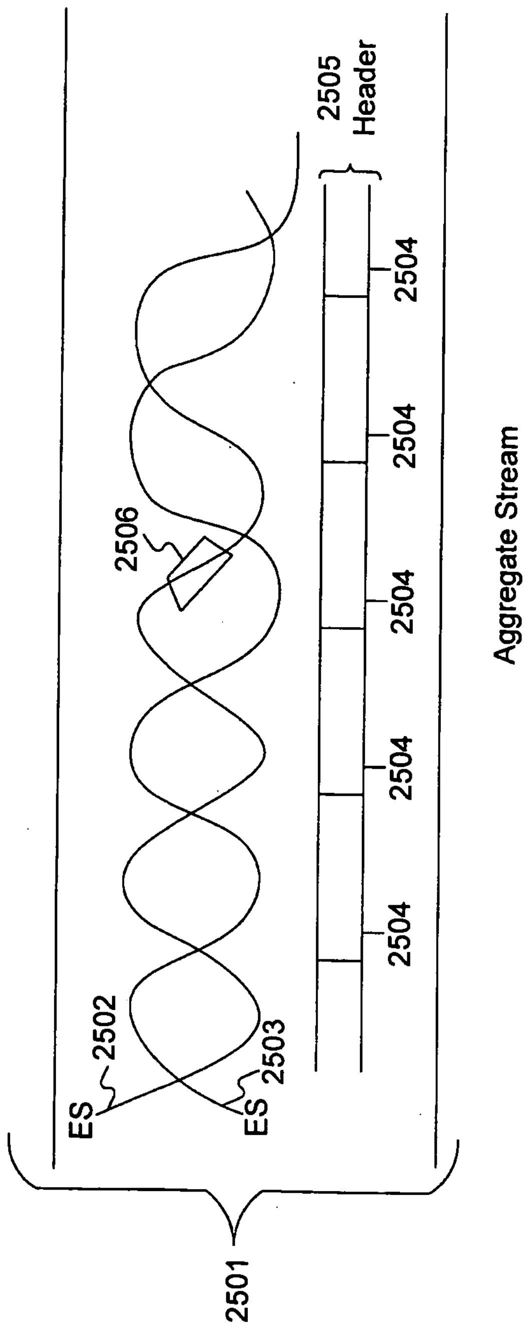

2505
Header
2504
2504
2504
2504
2504
2506
ES
2502
ES
2503
2501
Aggregate Stream

Fig. 25

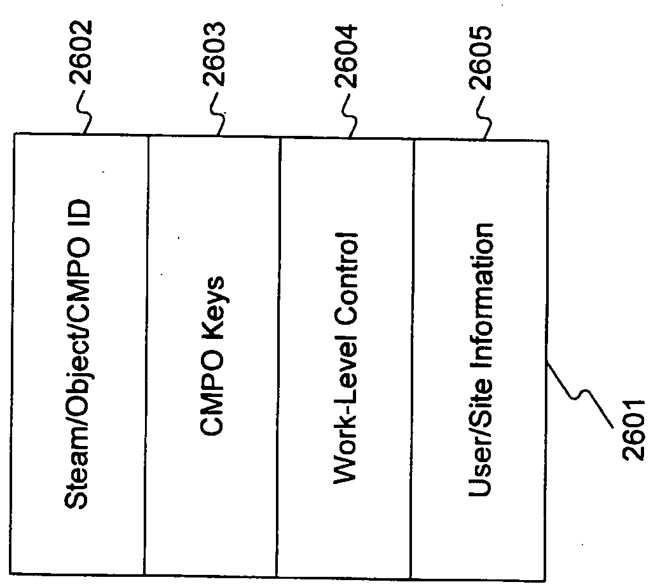

Steam/Object/CMPO ID
CMPO Keys
Work-Level Control
User/Site Information
2601
2602
2603
2604
2605

Fig. 26

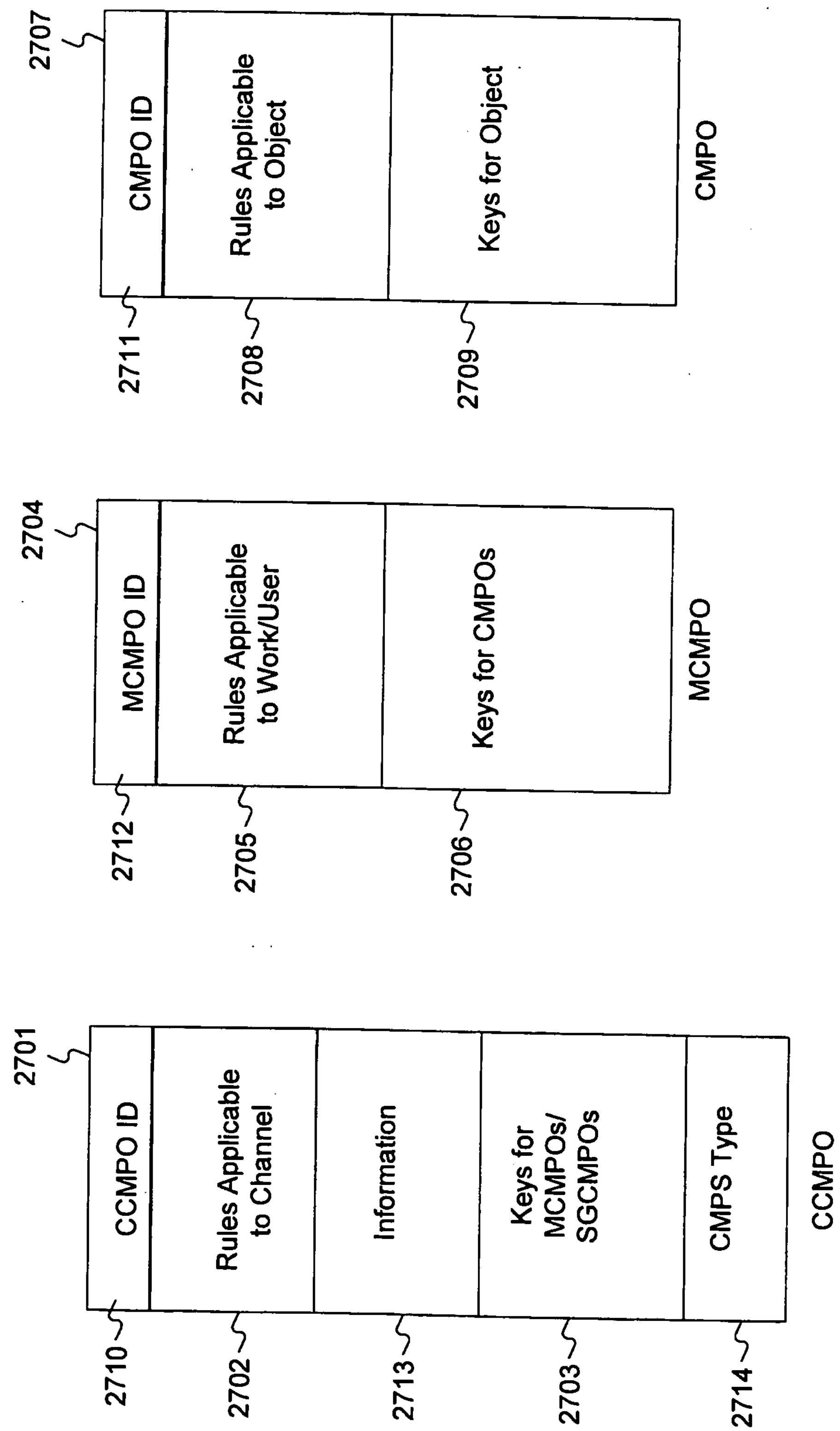

2707
CMPO ID
Rules Applicable to Object
Keys for Object
CMPO
2711
2708
2709
2704
MCMPO ID
Rules Applicable to Work/User
Keys for CMPOs
MCMPO
2712
2705
2706
2701
CCMPO ID
Rules Applicable to Channel
Information
Keys for MCMPOs/ SGCMPOs
CMPS Type
CCMPO
2710
2702
2713
2703
2714

Fig. 27

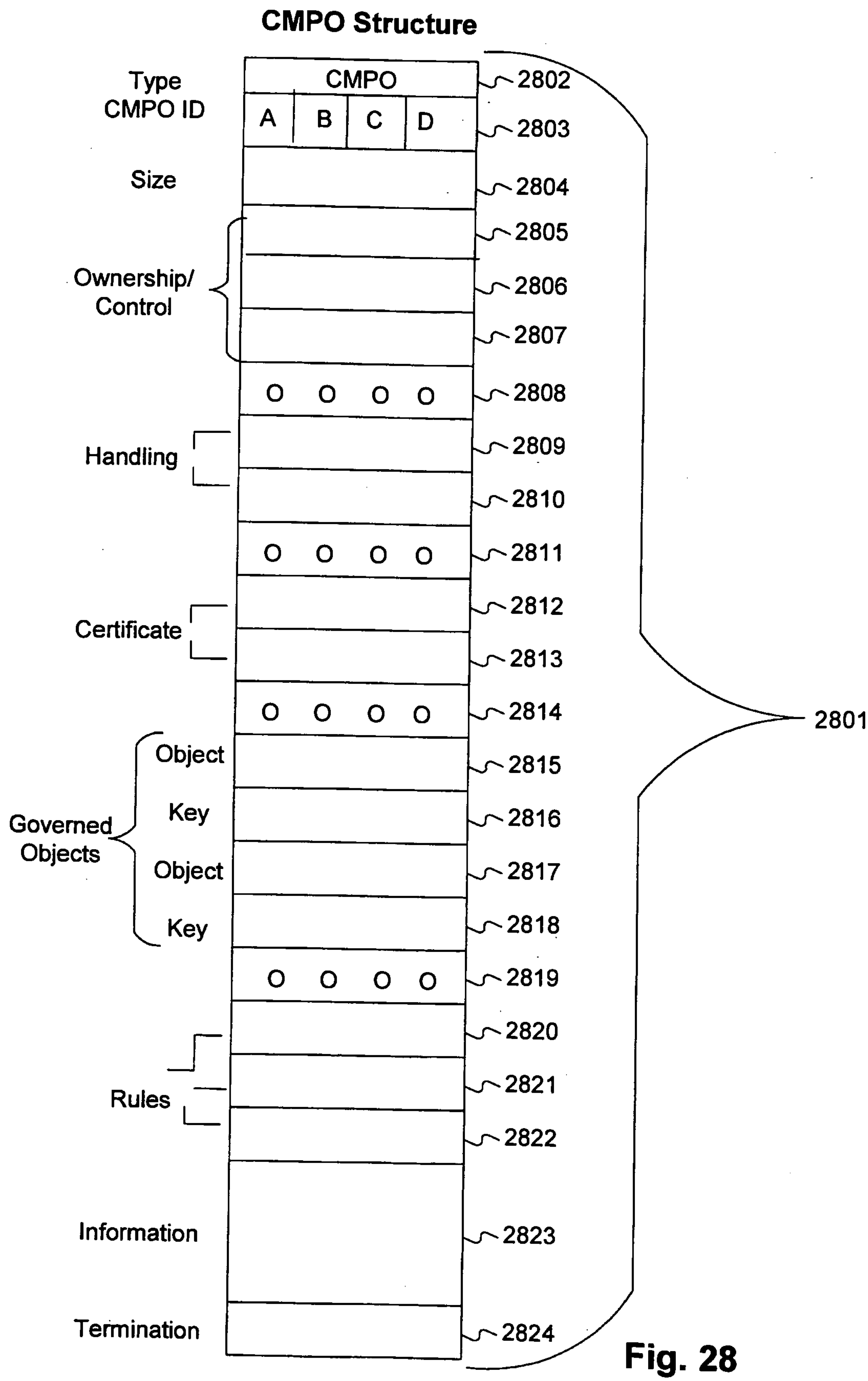

CMPO Structure
Type
CMPO ID
CMPO
A
B
C
D
Size
Ownership/
Control
Handling
Certificate
Governed
Objects
Object
Key
Object
Key
Rules
Information
Termination
O O O O
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2801

Fig. 28

INTERNATIONAL SEARCH REPORT

Intei ɔnal Application No
PCT/US 99/05734

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04N7/167 G06F1/00

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)
IPC 6 H04N G06F G11B

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 0 763 936 A (LG ELECTRONICS INC) 19 March 1997	1-4, 6-14, 17-20, 22-25
A	see abstract see column 6, line 27 - column 8, line 4 see column 9, line 6 - column 11, line 43 see column 16, line 47 - column 18, line 38 see figures 4,6A,6B,7 see figures 10,16 --- -/--	5,15,16, 21,26

[X] Further documents are listed in the continuation of box C.

[X] 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	Date of mailing of the international search report
1 July 1999	09/07/1999
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 Hampson, F

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/US 99/05734

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 714 204 A (LG ELECTRONICS INC) 29 May 1996	1-4, 6-14, 22-26
A	see abstract see page 6, line 14 - page 8, line 45 see figures 7,19A,B,20	5,15-21
A	EP 0 715 246 A (XEROX CORP) 5 June 1996 see abstract see page 3, line 46 - page 8, line 27 see figures 1-3,4A,4B	1-26
A	WO 97 25816 A (SONY CORP ;INOUE HAJIME (US); LEE CHUEN CHIEN (US); SONY ELECTRONI) 17 July 1997 see abstract see page 7, line 10 - page 10, line 7 see figures 2,3	1-26
A	EP 0 800 312 A (MATSUSHITA ELECTRIC IND CO LTD) 8 October 1997 see abstract see column 50, line 20 - column 51, line 53	1,6,7, 23-25

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte onal Application No
PCT/US 99/05734

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0763936 A	19-03-1997	CN 1150738 A	28-05-1997
		JP 9093561 A	04-04-1997
		US 5799081 A	25-08-1998
EP 0714204 A	29-05-1996	CN 1137723 A	11-12-1996
		JP 8242438 A	17-09-1996
		US 5757909 A	26-05-1998
EP 0715246 A	05-06-1996	US 5638443 A	10-06-1997
		JP 8263439 A	11-10-1996
WO 9725816 A	17-07-1997	AU 1344097 A	01-08-1997
		CN 1209247 A	24-02-1999
		EP 0882357 A	09-12-1998
		US 5889919 A	30-03-1999
EP 0800312 A	08-10-1997	WO 9714249 A	17-04-1997
		CN 1168054 A	17-12-1997
		EP 0789361 A	13-08-1997
		JP 10079174 A	24-03-1998

Form PCT/ISA/210 (patent family annex) (July 1992)

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 1/00 | A1 | (11) International Publication Number: WO 99/60461
(43) International Publication Date: 25 November 1999 (25.11.99)

(21) International Application Number: PCT/GB98/03828

(22) International Filing Date: 18 December 1998 (18.12.98)

(30) Priority Data:
09/080,030 15 May 1998 (15.05.98) US

(81) Designated States: CN, CZ, IL, IN, JP, KR, PL, SG, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(71) Applicant: INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(71) Applicant *(for MC only)*: IBM UNITED KINGDOM LIMITED [GB/GB]; North Harbour, Portsmouth, P.O. Box 41, Hampshire PO6 3AU (GB).

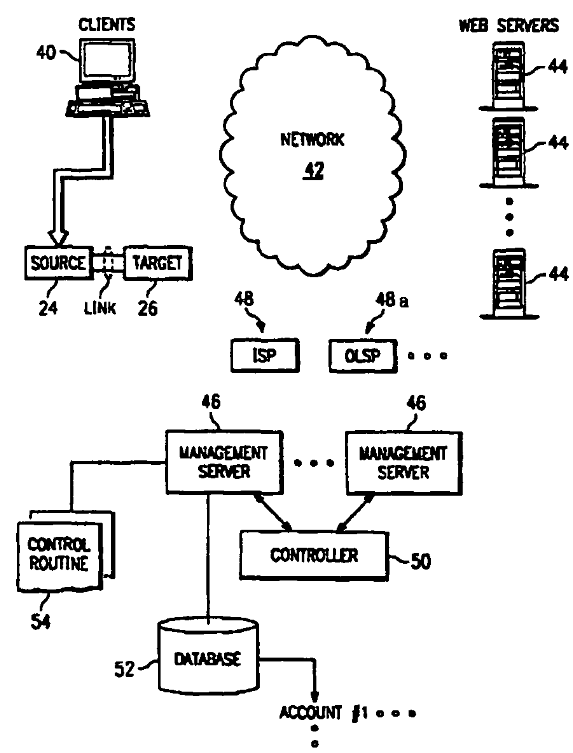
(72) Inventors: BERSTIS, Viktors; 5194 Cuesta Verde, Austin, TX 78746 (US). HIMMEL, Maria, Azua; 6403 Rain Creek Parkway, Austin, TX 78759 (US).

(74) Agent: BOYCE, Conor; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester, Hampshire SO21 2JN (GB).

(54) Title: ROYALTY COLLECTION METHOD AND SYSTEM FOR USE OF COPYRIGHTED DIGITAL MATERIALS ON THE INTERNET

(57) Abstract

A method, system and computer program product to facilitate royalty collection with respect to online distribution of electronically published material over a computer network. In one embodiment, a method for managing use of a digital file (that includes content subject to copyright protection on behalf of some content provider) begins by establishing a count of a number of permitted copies of the digital file. In response to a given protocol, a copy of the digital file is then selectively transferred from a source to a target. Thus, for example, the source and target may be located on the same computer with the source being a disk storage device and the target being a rendering device (e.g., a printer, a display, a sound card or the like). The method logs an indication each time the digital file is transferred from the source to a target rendering device, and the count is decremented upon each transfer. When the count reaches a given value (e.g., zero), the file is destroyed or otherwise prevented from being transferred from the source device. The indications logged are transferred to a management server to facilitate payment of royalties to the content provider.



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		

ROYALTY COLLECTION METHOD AND SYSTEM FOR USE OF COPYRIGHTED DIGITAL MATERIALS ON THE INTERNET

BACKGROUND OF THE INVENTION

Technical Field

The present invention relates generally to managing collection of royalties for electronically-published material distributed over a computer network.

Description of the Related Art

The World Wide Web is the Internet's multimedia information retrieval system. In the Web environment, client machines effect transactions to Web servers using the Hypertext Transfer Protocol (HTTP), which is a known application protocol providing users access to files (e.g., text, graphics, images, sound, video, etc.) using a standard page description language known as Hypertext Markup Language (HTML). HTML provides basic document formatting and allows the developer to specify "links" to other servers and files. In the Internet paradigm, a network path to a server is identified by a so-called Uniform Resource Locator (URL) having a special syntax for defining a network connection. Use of an HTML-compatible browser (e.g., Netscape Navigator or Microsoft Internet Explorer) at a client machine involves specification of a link via the URL. In response, the client makes a request to the server (sometimes referred to as a "Web site") identified in the link and, in return, receives in return a document or other object formatted according to HTML.

One of the technical advantages of the World Wide Web is the ease with which digital content (e.g., graphics, sound, video, movies and the like) may be transmitted and distributed to many users. Indeed, copying a digital file is as easy as clicking on a computer mouse. Copyright laws afford a copyright owner the exclusive right to reproduce the copyrighted work in copies, to distribute such copies, and to publicly perform and display the work. Each time a digital file is transferred over the Internet and copied onto a user s memory, the copyright owner's exclusive reproduction right is implicated (and possibly violated). Likewise, transmission of the copyrighted work over the physical wire is tantamount to a distribution. Indeed, in an open system (e.g., a personal computer accessing the World Wide Web through an Internet Service Provider (ISP)), copies of copyrighted materials can undergo unlimited further copying and transmission without the ability of the owner to collect appropriate compensation (e.g., royalties).

Many publishers or other content providers naturally are hesitant to make their copyrighted works available over the Internet due to the ease with which these materials may be copied and widely disseminated without adequate compensation. Presently, Internet commerce remains highly unregulated, and there is no central authority for managing collection and allocation of content provider royalties. Moreover, while publishers and content rights societies and organizations are attempting to address the legal and logistical issues, the art has yet to develop viable technical solutions.

One technique that has been proposed involves wrapping a copyrighted work in a copy protection "environment" to facilitate charging users for use of that information obtained from the Internet or World Wide Web. This approach, called COPINET, links a copyright protection mechanism with a copyright management system, and it is described in *Charging, paying and copyright - information access in open networks*, Bennett et al., 19th International Online Information Meeting Proceedings, Online Information 1995 pp. 13-23 (Learned Information Europe Ltd.). Publishers in such a system can determine an appropriate level of protection while monitoring use and managing the chain of rights. This approach is also said to provide protection for digital material even after delivery to the user workstation. In particular, copyright material is "wrapped" (by encryption) and "unwrapped" as a result of a specific authorization provided by a trusted subsystem. Material thus is only "visible" to the environment and thus any subsequent user actions, such as "save" or "copy", result in the protected material, or material derived from it, remaining in a protected state when outside the environment.

Although the above-described approach provides some advantages, it does not address the problem of managing the collection of royalties and/or the allocating of such payments to content providers. Moreover, it is not an accepting solution in the context of an open PC architecture such as implemented in the public Internet. It also requires the use of a separate trusted subsystem to generate the authorizations for particular content transfers, which is undesirable.

Other known techniques for managing use of content over the Internet typically involve electronic "wallets" or smart cards. Known prior art systems of this type are illustrated, for example, in U.S. Patent Nos. 5,590,197 and 5,613,001. These systems involve complex hardware and encryption schemes, which are expensive and difficult to implement in practice. They are not readily adaptable to provide general royalty payment schemes for Internet content usage.

Thus, there remains a need to provide improved methods and systems for collecting royalties on the Internet as a result of use of copyrighted content.

The present invention solves this important problem.

SUMMARY OF THE INVENTION

An object of this invention is to enable a pair of "certified" devices (e.g., a storage device and a rendering device) to operate within the context of a given security protocol and thereby manage copies of a digital file and associated copy control information.

Still another object of this invention is to enable a copyright proprietor to maintain a degree of control over copyrighted content even after that content has been fetched from a server and downloaded to a client machine, e.g., in a Web client-server environment.

A particular object of the present invention is to manage the number of copies of a digital file that may be made within a Web appliance having a secure disk storage and that is connectable to the Internet using a dialup network connection.

A still further object of this invention is to restrict a number of copies of a digital file that may be made at a given Web client machine connected to the World Wide Web.

It is yet another object of this invention to enable a publisher of an electronic document to control the number of copies of such document that may be made on the Internet by permitted users.

It is a more general object of this invention to manage permissible use of copyrighted content on the Internet and World Wide Web.

It is still another more general object of this invention to manage collection of information to facilitate payment of appropriate compensation to content providers and publishers arising from use of their copyrighted content on the Internet.

Another object of this invention is to manage the charging of users for information obtained from the Internet or World Wide Web.

A still further object of this invention is to facilitate royalty collection as a result of electronically published material distributed online over a computer network (e.g., the public Internet, an intranet, an extranet or other network).

One embodiment of the invention is a method for managing copies of a digital file, which includes content subject to copyright protection, on behalf of some content provider (e.g., an author, publisher or other). It is assumed that a given usage scheme has been established with respect to the file as defined in copy control information associated with the file. Thus, for example, the copy control information may define a set of payment options including, without limitation, prepayment (for "n" copies), pay-per-copy (as each copy is made), IOU (for copies made offline), or some other payment option. The copy control information may also include other data defining how the file is managed by the scheme including: a count of the number of permitted copies, a count of the number of permitted pay-per-copy versions, copyright management information, payee information, an expiration date (after which copying is no longer permitted), and the like.

The present invention assumes the existence of a pair of devices, a "source" and a "target", that have been or are certified to use the scheme. Typically, the "source" is a storage device while the "target" is a rendering device. An illustrative storage device may be disk storage, system memory, or the like. An illustrative rendering device may be a printer, a display, a sound card or the like. The source and target devices may both be storage devices (e.g., a Web server and a client disk storage). In either case, each of the devices comprising the pair is "certified" (typically upon manufacture) to operate under a given security protocol. Under the protocol, the devices include appropriate circuitry and/or software, as the case may be, to facilitate the establishment of a secure link between the storage and rendering devices. Each device requires the other to validate itself and thus prove that the device can be trusted to manage the content (namely, the digital file) sought to be protected.

When the technique is implemented in an "open" client-server environment, hardware devices (e.g., microcontrollers) preferably are used in the storage and rendering devices to facilitate generation of the secure link. When the technique is implemented in a "closed" Web appliance environment, the secure link may be established and managed using software resident in the control routines associated with the storage and rendering devices. The secure link may be established and managed in software under such conditions because, in the Web appliance environment, it is possible to readily disable the secure link in the event of tampering with the appliance housing or other circuitry. Regardless of the environment, the secure link is first established between the "certified" storage and rendering devices. Thereafter, the digital file, together with at least part of its copy control information, is transferable between the storage and rendering devices in accordance with the particular usage and payment scheme being utilized. Thus, for example, if a prepayment scheme is implemented and an expiration date (associated therewith) has not occurred, a given number of copies of the file may be transferred between the storage and rendering devices. The prepayment funds are collected at a central location and then redistributed to the copyright proprietor or some third party.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a representative system in which the present invention is implemented;

Figure 2 is a simplified block diagram of a source device and a target device connected by a channel over which a digital file is transferred according to the present invention;

Figure 3 is an illustrative example of a source device connected to a set of target rendering devices in a client computer;

Figure 4 is a block diagram of a representative copyright management system according to the present invention;

Figure 5 is a flowchart of a preferred method of managing a digital file according to the present invention;

Figure 6A is pictorial representation of a data processing system unit connected to a conventional television set to form a "Web" appliance;

Figure 6B is a pictorial representation of a front panel of the data processing system unit;

Figure 6C is a pictorial representation of a rear panel of the data processing system unit;

Figure 6D is a pictorial representation of a remote control unit associated with the data processing system unit; and

Figure 7 is a block diagram of the major components of the data processing system unit.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A representative system in which the present invention is implemented is illustrated in **Figure 1**. A plurality of Internet client machines **10** are connectable to a computer network Internet Service Provider (ISP) **12** via a "resource" such as a dialup telephone network **14**. As is well known, the a dialup telephone network usually has a given, limited number of connections **16a-16n**. ISP **12** interfaces the client machines **10** to the remainder of the network **18**, which includes a plurality of Internet server machines **20**. A client machine typically includes a suite of known Internet tools (e.g., Web browser **13**) to access the servers of the network and thus obtain certain services. These services include one-to-one messaging (e-mail), one-to-many messaging (bulletin board), on-line chat, file transfer and browsing. Various known Internet protocols are used for these services. Thus, for example, browsing is effected using the Hypertext Transfer Protocol (HTTP), which provides users access to multimedia files using Hypertext Markup Language (HTML). The collection of servers that use HTTP comprise the World Wide Web, which is the Internet's multimedia information retrieval system.

As will be described in more detail below, the present invention may be implemented in hardware and/or in software. The software implementation is particularly useful when the client machine is an Internet or Web appliance, such as illustrated in **Figures 6A-6D**. In the case of the software implementation, a client machine has associated therewith a software routine **15** designed to perform one or more of the functions of the digital file copy protection method, as will be described. The software is preferably a client application (although it may be implemented with the browser as a plug-in, or with a client-side proxy, or as a standalone application). Alternatively, the agent is built into the browser, or it is implemented as a Java applet or standalone application. Thus, as used herein, in this particular embodiment, the software **15** is any application running on a client machine **10** that performs the copy protection/royalty management task(s) on behalf of the user(s) of that client according to the present invention.

The discussion which follows primarily uses the words "copying" or "copies" to describe the control of the further exercise of a copyright right for a particular work. The reader should understand that "copying" could include other types of rendering of the work for different devices. That is, "copying" in a printer would entail printing on paper or another substrate. Copying on a display is presenting an image on the screen. Copying in an audio device would be the performance of an audio portion of the work. Each of these devices both storage devices, e.g., hard disks, tapes in CDR, and rendering devices, e.g., prints, display graph, audio player, movie player, should be equipped with the present invention so that the copies are controlled throughout the systems and networks until their final rendering place.

The present invention is a method for managing copies of a digital file, which includes content subject to copyright protection, on behalf of some content provider (e.g., an author, publisher or other). It is assumed that a given payment scheme has been established with respect to the file. Thus, for example, such payment schemes include, without limitation, prepayment (for "n" copies), pay-per-copy (as each copy is made), IOU (for copies made offline), or some other payment option. In a prepayment option, a user prepays funds for the right to obtain copies of the digital file. In a pay-per-copy (or "pay as you go") option, the user pays for each copy of the digital file when the file is copied. In an IOU scheme, the user makes copies of the digital file (e.g., while the client machine is not connected to the network) and generates an IOU (or many IOUs) that are then submitted to a clearinghouse or other payment entity when the user later goes online. Other payment schemes (such as a combination of the above options) may also be implemented.

The payment scheme is preferably defined in copy control information associated with the file and established by the author, publisher or some other third party. Thus, for example, the copy control information may also include a count of the number of permitted copies, a count of the number of permitted pay-per-copy versions, a count of the number of copies that may be made under an IOU payment option, copyright management information identifying the author, publisher and/or other license or use restrictions, information about a bank or other financial institution that handles use payments and their reconciliation, one or more expiration dates (after which copying is no longer permitted), and the like.

The copy control information associated with a given file thus defines a usage scheme for the file because it includes information that controls how the content may be used, how such use is paid for, over what period the content may be used, and other such information. A particular usage scheme (or some portion thereof) may also be implemented in the devices between which the file is transferred, although preferably such restrictions are defined by the content provider.

According to the present invention as illustrated in **Figure 2**, the present invention assumes the existence of a pair of devices, a "source" **24** and a "target" **26**, that have been or are certified to use the scheme. In particular, devices that implement the inventive scheme preferably include a device certificate that is not accessible (and thus is free from tampering) and stored therein. The certificate evidences that the device is capable of understanding a given security protocol useful in carrying out the protection scheme. A representative security protocol is CSS, or the Content Scrambling System protocol, available commercially from Matsushita Corp. Thus, for example, if the source device is a disk storage, the device certificate is typically stored inside a secure chip within the device control hardware. Typically, each of the devices is "certified" upon manufacture, although this is not a requirement.

As also illustrated in **Figure 2**, a channel **28** is established between the source and target devices over which copies of a digital file (that is subject to the scheme) are communicated in a secure fashion. Thus, prior to transfer of the digital file, the channel **28** is first established between the devices to ensure that the copy restrictions (such as set forth in the copy control information) may be enforced. Typically, this is accomplished by having each device (in accordance with the security protocol implemented) require the other device (of the pair) to verify that its device certificate is valid. An appropriate message exchange may be used for this purpose as defined in the protocol. Once the secure link has been established, each of the devices can be trusted to control the digital file in accordance with the file's copy control information.

Typically, the "source" **24** is a storage device while the "target" **26** is a rendering device. An illustrative storage device may be disk storage, system memory, or the like. An illustrative rendering device may be a printer, a display, a sound card or the like. The source and target devices may both be storage devices (e.g., a Web server and a client disk storage).

When the technique is implemented in an "open" client-server environment, hardware devices (e.g., microcontrollers) are used in the storage and rendering devices to facilitate generation and management of the secure link. When less security may be tolerated, some of these functions may be implemented in software. When the technique is implemented in a "closed" Web appliance environment (**Figures 6A-6D**), the secure link may be established in whole or in part using software resident in the control routines associated with the storage and rendering devices. The secure link may be established in software under such conditions because, in the Web appliance environment, it is possible to readily disable the secure link in the event of tampering with the appliance housing or other circuitry. Regardless of the environment, the secure link is first established between the "certified" storage and rendering devices. Thereafter, the digital file, together with at least part of its copy control information, is transferable between the storage and rendering devices in accordance with the particular usage scheme defined, for example, by the copy control information. Thus, for example, if a prepayment scheme is implemented and an expiration date (associated therewith) has not occurred, a given number of copies of the file may be transferred between the storage and rendering devices.

Thus, as illustrated in **Figure 2** in simplified form, the digital file copy protection method and system of the present invention involves a "source" device **24** (or one or more of such devices), and a set of one or more "target" devices **26**a-n connected via the secure channel or link **28**. The physical characteristics of the channel, of course, depend on whether the source and target devices are located in the same machine or are in separate machines connected via a network. In a network connection, the link may be a conventional TCP/IP connection. Channel **28** may be a physically secure channel (such as a https connection), but this is not required as the given security protocol in the certified devices establishes a secure link. According to the invention, once the link is established, one or more digital files are transferred (under the control of a control routine or mechanism) between the certified devices in an predictable, auditable manner so that (a) a controlled number of file transfers can be made, and (b) the precise number of file transfers (and their particular use) may be readily documented to facilitate dissemination of royalties or some such other consideration, typically to providers of such content. Generalizing, prior to transfer of a given digital file (or set of files, or file component) from the source to the target via the secure link, that transfer must first be authorized, and the transfer itself is then capable of being associated with some royalty payment then due to a content provider for use of such file. The scheme thus facilitates implementation of a generalized copyright management/royalty collection and distribution scheme.

As previously mentioned, the source **24** and target **26** may be located on the same computer. **Figure 3** illustrates this particular connection for a disk storage subsystem **24'** and the target rendering devices, namely printer **26a'**, display **26b'** and sound card **26c'**. The illustrated computer is a Web appliance, in which case the secure link may be established (as noted above) using software. Thus, in this example, each source and/or target device includes appropriate control software (part of software **15** as described above) to facilitate creation of the secure channel. Although not meant to be limiting, one convenient mechanism to create the channel involves each of the devices to generate a random number **30**, which numbers are then supplied to a key generation algorithm **32** in a known manner to generate a secret of "private" key **34**. The key **34** may be generated for each digital file to be transferred over the link **28**, or a signal key may be used for a set of such files, or even for a particular browsing session. To create the secure channel, the software resident on the disk storage encrypts the digital file as it leaves the source device.

The target device then decrypts the digital file using the key prior to rendering. In this way, the digital file cannot be readily intercepted as it is being transferred between these devices. As noted above, each of the source and target devices may also include secure chips or other known hardware devices to facilitate or augment such secure transfer of the digital file between the devices.

The particular mechanism for securing the channel between the source and target may be quite varied, and the present invention contemplates the use of any now known or later-developed technique, system or method for securing such communications. Thus, for example, another technique that may be used would be a public key cryptosystem.

Figure 4 is a block diagram illustrating a representative copyright royalty management system implemented according to the present invention. In this system, it is assumed that client computers **40** access the computer network **42** (e.g., the public Internet, an intranet, an extranet, or other computer network) to obtain access to Web-like documents supported on Web servers **44**. One or more management servers **46** are connectable to the system via an access provider **48**, and a control management server **50** may be used to facilitate scaling of the architecture if required. Control management server **50** may be controlled by a regulatory or rights agency that has responsibility for managing collection and distribution of copyright royalties.

A given management server includes a database **52** and appropriate control routines **54** for establishing a royalty account **55** for content providers. It is envisioned (although not required) that given content providers will subscribe to a royalty collection service implemented by the present invention and perhaps pay a fee (e.g., a commission or service charge) for the service provided. A given content provider thus may subscribe to the service to receive royalty payments for the use of his or her copyrighted content by users of the client machines. To this end, control routines **54** are used to establish an account for each of a set of given content providers, with each account including a representation of a given royalty value (which may be $0 when the account is established). A control routine then adjusts the given royalty value in a given provider account in response to receipt of an indication that a given digital file associated with the given content provider has been transferred from a source **24** to a target rendering device **26** in a given client computer **40**. Periodically, the content provider account is adjusted for any service or processing fees, and the remainder of the account is then distributed to the content provider. In the situation where the content provider is willing to allow his or her content (a given digital file) to be used with charges for such use paid later, a given bit may be set in the file's copy control information indicating such preference. Other data in the copy control information may be used to set or control other content provider preferences with respect to use of the file within the context of the inventive scheme.

Figure 5 is a flowchart of one method of managing royalty account collection with respect to a particular digital file when a prepayment option is utilized. In this representative example, the digital file is an image (i.e. a .jpeg file) having a copyright owned by a given content proprietor or provider. Of course, the principles of the present invention are designed to be implemented collectively with many such digital files, and the following description is thus merely representative of one type of basic payment scheme. The routine assumes initially that a usage or payment account has been established for a given client computer (or a user of that computer). This is step **60** in the flowchart. It is also assumed that a royalty account has been established for the content provider at one of the management servers as previously described. This is step **62** in the flowchart. One of ordinary skill will appreciate that steps **60** and **62** need not be in any particular sequence. Step **60** typically involves the user prepaying some amount of funds into an account from which payments may be withdrawn, although this is not required.

At step **64**, a count is established by a control routine for the particular digital file. Typically, this is a count of a number of permitted copies of the digital file that may be transferred from the source to one or more target devices according to the present invention. This number, as noted above, is typically identified in the file's copy control information. The count is usually a positive integer, which is then decremented (by the control routine) down to zero as permitted or authorized copies are made. Alternatively, of course, the count may begin at zero (or any other arbitrary number), which is then incremented (by the control routine) to the threshold value identified in the copy count information. As noted above, the count may be set by the copyright proprietor, by a system operator, by a Webmaster, by hardware constraints, or by any other party or entity having authority and/or ability to set the count. Under certain circumstances, e.g., where a prepaid user account is used, it may be unnecessary to use an explicit count as the number of copies transferred may simply depend on the royalty assessed per copy. Thus, the "count" as used herein may be expressed explicitly or implicitly. The digital file may be stored on the client already, or it may be available from a Web server or other storage or archive. The particular location from which the digital file is sourced initially does not matter. Step **64** assumes, however, that the image is located already at the source device. If the file is not present at the source, it may be necessary to obtain it (although, conceptually, the "source" may be broadly construed as the original or initial location of the file).

At step **66**, a test is done repeatedly to determine whether a request for the image has been received. If not, the routine cycles on step **66** and waits for such a request. If the outcome of the test at step **66** is positive, then the routine continues at step **68** by testing whether the given client computer (which generated the request) is authorized to effect the transfer. Step **68** may comprise a simple comparison of the user's account balance and the royalty amount to be assessed. If the user's account balance is large enough, the transfer may be allowed. Or, step **68** may simply test whether the count has a value indicating that further copies may be made. More typically, step **68** will require that the count be non-zero (in the situation where the count is positive and decremented to zero) and the user have sufficient funds allocated to pay the royalty assessment for use of the image. The step **68** may also test whether a given expiration date set in the copy count information has past.

If the outcome of the test at step **68** is negative, the transfer is not authorized, and the routine branches to step **70** to so notify the user of the client machine. Such notification may be in the form of an error or "access denied" message or the like. The user may be informed merely that a preset expiration date has passed or that his or her prepaid account is exhausted and requires more funds. If, however, the outcome of the test at step **68** is positive, the digital file may be transferred to the target. The routine then branches to step **72** to initiate the copy transfer. Preferably, all bytes of the file must be transferred before the transfer is considered valid. At step **74**, the control routine count is adjusted (e.g., decremented) and/or a given charge is allocated against the user's account. The given charge may be equal to the royalty or use charge, or some fixed percentage thereof (e.g., **105%**) reflecting that royalty plus some service charge). At step **76**, the appropriate content provider account is adjusted by the amount of the royalty payment (plus or minus appropriate service fees or other charges).

Neither step **74** nor step **76** need occur at the time of the file transfer. Typically, the account adjustments will take place in batch at a given time. Thus, for example, where the Web client is a Web appliance connected to the computer network via a dialup connection, the account information may be transferred to the management server upon establishing a given connection (e.g. perhaps once each day). Other variations regarding the timing of delivery of this information are, of course, within the scope of the present invention.

The present invention thus provides numerous advantages. Certified source and target devices first establish a secure link between themselves. Upon transfer of the file copy between source and target, the control routine records an appropriate indication thereof in the copy count, and the central authority is notified of the transfer of the digital file. Such notification may occur upon transfer of the digital file between the source and target devices, or at some later time (e.g., upon dialup connection of the computer to the network). Royalty accounts are then managed at a central authority; to facilitate distribution of royalties to content owners/publishers. When the copy count reaches the authorized limit (as set in the copy control information), the control routine destroys the file or otherwise prevents further copying of the digital file.

Thus, in one embodiment, the user establishes a "prepaid" account from which royalty or usage payments are drawn against as files are copied/transmitted. The system detects use of the file and, preferably, allows only a certain number of copies of the file to be made before the document is destroyed or otherwise rendered inaccessible (from the client machine). The resulting copyright management infrastructure is robust, secure, scaleable and easily managed.

In one embodiment of this invention as described above, the Internet client is a data processing system or a so-called "Web appliance" such as illustrated in **Figures 6A-6D** and 7. **Figure 6A** is a pictorial representation of the data processing system as a whole. Data processing system **100** in the depicted example provides, with minimal economic costs for hardware to the user, access to the Internet. Data processing system **100** includes a data processing unit **102**. Data processing unit **102** is preferably sized to fit in typical entertainment centers and provides all required functionality, which is conventionally found in personal computers, to enable a user to "browse" the Internet. Additionally, data processing unit **102** may provide other common functions such as serving as an answering machine or receiving facsimile transmissions.

Data processing unit **102** is connected to television **104** for display of graphical information. Television **104** may be any suitable television, although color televisions with an S-Video input will provide better presentations of the graphical information. Data processing unit **102** may be connected to television **104** through a standard coaxial cable connection. A remote control unit **106** allows a user to interact with and control data processing unit **102**. Remote control unit **106** allows a user to interact with and control data processing unit **102**. Remote control unit **106** emits infrared (IR) signals, preferably modulated at a different frequency than the normal television, stereo, and VCR infrared remote control frequencies in order to avoid interference. Remote control unit **106** provides the functionality of a pointing device (such as a mouse, glidepoint, trackball or the like) in conventional personal computers, including the ability to move a cursor on a display and select items.

Figure 6B is a pictorial representation of the front panel of data processing unit **102**. The front panel includes an infrared window **108** for receiving signals from remote control unit **106** and for transmitting infrared signals. Data processing unit **102** may transmit infrared signals to be reflected off objects or surfaces, allowing data processing unit **102** to automatically control television **104** and other infrared remote controlled devices. Volume control **110** permits adjustment of the sound level emanating from a speaker within data processing unit **102** or from television **104**. A plurality of light-emitting diode (LED) indicators **112** provide an indication to the user of when data processing unit **102** is on, whether the user has messages, whether the modem/phone line is in use, or whether data processing unit **102** requires service.

Figure 6C is a pictorial representation of the rear panel of data processing unit **102**. A three wire (ground included) insulated power cord **114** passes through the rear panel. Standard telephone jacks **116** and **118** on the rear panel provide an input to a modem from the phone line and an output to a handset (not shown). The real panel also provides a standard computer keyboard connection **120**, mouse port **122**, computer monitor port **124**, printer port **126**, and an additional serial port **128**. These connections may be employed to allow data processing unit **102** to operate in the manner of a conventional personal computer. Game port **130** on the rear panel provides a connection for a joystick or other gaming control device (glove, etc.). Infrared extension jack **132** allows a cabled infrared LED to be utilized to transmit infrared signals. Microphone jack **134** allows an external microphone to be connected to data processing unit **102**.

Video connection **136**, a standard coaxial cable connector, connects to the video-in terminal of television **104** or a video cassette recorder (not shown). Left and right audio jacks **138** connect to the corresponding audio-in connectors on television **104** or to a stereo (not shown). If the user has S-Video input, then S-Video connection **140** may be used to connect to television **104** to provide a better picture than the composite signal. If television **104** has no video inputs, an external channel 3/4 modulator (not shown) may be connected in-line with the antenna connection.

Figure 6D is a pictorial representation of remote control unit **106**. Similar to a standard telephone keypad, remote control unit **106** includes buttons **142** for Arabic numerals **0** through **9**, the asterisk or "star" symbol

(*), and the pound sign (#). Remote control unit also includes "TV" button **144** for selectively viewing television broadcasts and "Web" button **146** for initiating "browsing" of the Internet. Pressing "Web" button **146** will cause data processing unit **102** to initiate modem dial-up of the user's Internet service provider and display the start-up screen for an Internet browser.

A pointing device **147**, which is preferably a trackpoint or "button" pointing device, is included on remote control unit **106** and allows a user to manipulate a cursor on the display of television **104**. "Go" and "Back" buttons **148** and **150**, respectively, allow a user to select an option or return to a previous selection. "Help" button **151** causes context-sensitive help to be displayed or otherwise provided. "Menu" button **152** causes a context-sensitive menu of options to be displayed, and "Update" button **153** will update the options displayed based on the user's input, while home button **154** allows the user to return to a default display of options. "PgUp" and "PgDn" buttons **156** and **158** allows the user to change the context of the display in display-sized blocks rather than by scrolling. The message button **160** allows the user to retrieve messages.

In addition to, or in lieu of, remote control unit **106**, an infrared keyboard (not shown) with an integral pointing device may be used to control data processing unit **102**. The integral pointing device is preferably a trackpoint or button type of pointing device. A wired keyboard (also not shown) may also be used through keyboard connection **120**, and a wired pointing device such as a mouse or trackball may be used through mouse port **122**. When a user has one or more of the remote control unit **106**, infrared keyboard, wired keyboard and/or wired pointing device operable, the active device locks out all others until a prescribed period of inactivity has passed.

Referring now to **Figure 7**, a block diagram for the major components of data processing unit **102** is portrayed. As with conventional personal computers, data processing unit **102** includes a motherboard **202** containing a processor **204** and memory **206** connected to system bus **280**. Processor **205** is preferably at least a **486** class processor operating at or above **100** MHz. Memory **206** may include cache memory and/or video RAM. Processor **205**, memory **206**, and system bus **208** operate in the same manner as corresponding components in a conventional data processing system.

Video/TV converter **210**, located on motherboard **202** and connected to system bus **208**, generates computer video signals for computer monitors, a composite television signal, and an S-Video signal. The functionality of Video/TV converter **210** may be achieved through a Trident **TVG9685** video chip in conjunction with an Analog Devices AD722 converter chip. Video/TV converter **210** may require loading of special operating system device drivers.

Keyboard/remote control interface unit **212** on motherboard **202** receives keyboard codes through controller **214**, regardless of whether a wired keyboard/pointing device or an infrared keyboard/remote control is being employed. Infrared remote control unit **106** transmits signals which are ultimately sent to the serial port as control signals generated by conventional mouse or pointing device movements. Two buttons on remote control unit **106** are interpreted identically to the two buttons on a conventional mouse, while the remainder of the buttons transmit signals corresponding to keystrokes on an infrared keyboard. Thus, remote control unit **106** has a subset of the function provided by an infrared keyboard.

Connectors/indicators **216** on motherboard **202** provide some of the connections and indicators on data processing unit **102** described above. Other connections are associated with and found on other components. For example, telephone jacks **116** and **118** are located on modem **222**. The power indicator within connectors/indicators **216** is controlled by controller **214**.

External to motherboard **202** in the depicted example are power supply **218**, hard drive **220**, modem **222** and speaker **224**. Power supply **218** is a conventional power supply except that it receives a control signal from controller **214** which effects shut down of all power to motherboard **202**, hard drive **220** and modem **222**. Power supply **218**, in response to a signal from controller **214**, is capable of powering down and restarting data processing unit **102**.

Controller **214** is preferably one or more of the 805x family controllers. Controller **214** receives and processes input from infrared remote control **106**, infrared keyboard, wired keyboard, or wired mouse. When one keyboard or pointing device is used, all others are locked out (ignored) until none have been active for a prescribed period. Then the first keyboard or pointing device to generate activity locks out all others. Controller **214** also directly controls all LED indicators except that indicating modem use. As part of the failure recovery system, controller **214** specifies the boot sector selection during any power off-on cycle.

Hard drive **220** contains operating system and applications software for data processing unit **102**, which preferably includes IBM DOS 7.0, a product of International Business Machines Corporation in Armonk, New York; an operating system **221** such as Windows 3.1 (or higher), a product of Microsoft Corporation in Redmond, Washington; and a browser **223** such as Netscape Navigator (Version 1.0 or higher), a product of Netscape Communications Corporation in Mountain View, California. Hard drive **220** may also support an SMTP mechanism to provide electronic mail, an FTP mechanism to facilitate file transfers from Internet FTP sites, and other Internet protocol mechanisms, all in a known manner. Hard drive **220** is not generally accessible to the user of the Web appliance.

Modem **222** may be any suitable modem used in conventional data processing systems, but is preferably a 33.6 kbps modem supporting the V.42bis, V.34, V.17 Fax, MNP 1-5, and AT command sets. Modem **222** is connected to a physical communication link **227**, which, in turn, in connected or connectable to the Internet (not shown).

Those skilled in the art will recognize that the components depicted in **Figures 6A-6D** and **7** and described above may be varied for specific applications or embodiments. Such variations in which the present invention may be implemented are considered to be within the spirit and scope of the present invention.

According to the invention, the client machine (typically the hard drive **220**) also includes a proxy **225**. Preferably, the proxy is implemented in software and includes a cache **227** associated therewith. The cache may be integral to the proxy or logically associated therewith. The cache preferably has a size up to several hundred megabytes, which is substantially larger than the standard cache associated with a browser such as Netscape Navigator. The client machine also includes a protocol stack **229** (e.g., a TCP/IP protocol stack) and a sockets mechanism **231**, which are used to support communications in a known manner. According to the invention, the proxy **225** is advantageously located on the client along with the browser. Thus, the proxy is sometimes referred to as a "client side" proxy.

Preferably, the proxy starts up when the Web appliance is booted up. Connectivity between the proxy and the browser is achieved using the sockets mechanism by configuring the browser to pass the HTTP requests to the proxy. To send an HTTP GET request, the browser creates a packet (including the URL and other information) and then opens a socket using the sockets mechanism. The packet is then sent to the IP address/port number to service the HTTP request. Thus, when the browser issues an HTTP GET request, it binds to the socket and sends the request. The request is then intercepted and processed by the proxy instead of being sent directly over the network, all in the manner previously described.

Although in the preferred embodiment the client machine is a Web "appliance", this is not a requirement of the present invention. Thus, a client machine **10** may be a personal computer such as a desktop of notebook computer, e.g., an IBM® or IBM-compatible machine running under the OS/2® operating system, an IBM ThinkPad® machine, or some other Intel x86 or Pentium®-based computer running Windows 95 (or the like) operating system.

A representative server platform comprises an IBM RISC System/6000 computer (a reduced instruction set of so-called RISC-based workstation) running the AIX (Advanced Interactive Executive Version 4.1 and above) Operating System **21** and Server program(s) **22**. The platform **20** also includes a graphical user interface (GUI) **23** for management and administration. It may also include an application programming interface (API) **24**. HTTP GET requests are transferred from the client machine to the server platform, typically via the dial-up computer network, to obtain documents or objects formatted according to HTML or some other markup language. While the above platform is useful, any other suitable hardware/operating system/server software may be used.

One of the preferred implementations of the client side or server side mechanisms of the invention is as a set of instructions (program code) in a code module resident in the random access memory of the computer. Until required by the computer, the set of instructions may be stored in another computer memory, for example, in a hard disk drive, or in a removable memory such as an optical disk (for eventual use in a CD ROM) or floppy disk (for eventual use in a floppy disk drive), or downloaded via the Internet or other computer network.

In addition, although the various methods described are conveniently implemented in a general purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps.

As used herein, "Web client" should be broadly construed to mean any computer or component thereof directly or indirectly connected or connectable in any known or later-developed manner to a computer network, such as the Internet. The term "Web server" should also be broadly construed to mean a computer, computer platform, an adjunct to a computer or platform, or any component thereof. Of course, a "client" should be broadly construed to mean one who requests or gets the file, and "server" is the entity which downloads the file. Moreover, although the present invention is described in the context of the Hypertext Markup Language

(HTML), those of ordinary skill in the art will appreciate that the invention is applicable to alternative markup languages including, without limitation, SGML (Standard Generalized Markup Language) and XML (Extended Markup Language).

In addition, the term "Web appliance" should be broadly construed to cover the display system illustrated in **Figures 6A-6D**, as well as any other machine in which a browser application is associated with some television class or other display monitor. Moreover, while the preferred embodiment is illustrated in the context of a dial-up network, this is not a limitation of the present invention. There may be other "bottleneck" resources in a direct connect network that could be managed indirectly by using this approach.

CLAIMS

1. A method for managing use of a digital file, comprising the steps of:

establishing a secure link between a pair of devices, each of the devices being certified to operate under a given security protocol;

establishing a usage scheme defining one or more conditions under which the digital file may be transferred between the pair of devices; and

transferring one or more copies of the digital file over the secure link between the pair of devices in accordance with the established usage scheme.

2. The method as described in Claim 1 wherein the pair of devices include a storage device and a rendering device.

3. The method as described in Claim 2 wherein the storage device and the rendering device are located in a computer.

4. The method as described in Claim 2 wherein the storage device is located in a first computer and the rendering device is located in a second computer and the secure link is established over a computer network connecting the first and second computers.

5. The method as described in Claim 4 wherein the second computer is a personal computer and the rendering device includes circuitry for establishing the secure link.

6. The method as described in Claim 4 wherein the second computer is a Web appliance and the rendering device includes software for establishing the secure link.

7. The method as described in Claim 2 wherein the rendering device is selected from a group of rendering devices consisting essentially of a printer, a display, and a sound card.

8. The method as described in Claim 1 further including the step of establishing an account representing a given monetary value.

9. The method as described in Claim 8 further including the step of allocating a given charge against the given monetary value when a copy of the digital file is transferred between the pair of devices.

10. The method as described in Claim 9 further including the step of associating the given charge with a content provider account to facilitate the payment of the given consideration to the provider of the digital file.

11. The method as described in Claim 1 wherein the usage scheme includes a given payment method.

12. A method for managing use of digital material in a computer network, comprising the steps of:

establishing an account for a given client computer including a representation of a given monetary value;

establishing an account for a given content provider including a representation of a given royalty value;

establishing a count of a number of permitted copies of a digital file;

in response to a given protocol, transferring a copy of the digital file from a source to a target associated with the given client computer;

adjusting the given monetary value in the account of the given client computer; and

adjusting the given royalty value in the account of the given content provider.

13. The method as described in Claim 12 wherein the given protocol includes the steps of:

determining whether a given client computer requesting transfer of the digital file is authorized to effect the transfer;

if the client is authorized to effect the transfer of the digital file, determining whether the count has a given value; and

if the count has the given value, transferring the digital file from the source to the target.

14. The method as described in Claim 13 wherein the given value is a non-zero value.

15. The method as described in Claim 13 wherein the given protocol further includes the step of adjusting the count after a copy of the digital file has been transferred.

16. The method as described in Claim 15 wherein the count is decremented.

17. The method as described in Claim 12 wherein the source and target are located in the given client computer connected to the computer network.

18. The method as described in Claim 17 wherein the source is a disk storage device and the target is a device selected from a group of rendering devices consisting essentially of a printer, a display, and a sound card.

19. The method as described in Claim 12 wherein the source is located on a first computer and the target is located on a second computer connected to the first computer via the computer network.

20. A method for managing use of digital material in a computer network including a Web client connectable to a Web server, comprising the steps of:

establishing a count of a number of permitted copies of a digital file located at a source device in the Web client;

in response to a given protocol, transferring one or more copies of the digital file from the source device to a set of one or more target rendering devices in the Web client; and

for each such transfer from the source device to one of the target rendering devices, logging an indication that the digital file has been transferred to facilitate payment of a given consideration to a provider of the digital file.

21. The method as described in Claim 20 wherein the Web client is a Web appliance and the source device is a secure disk storage.

22. The method as described in Claim 21 wherein each target rendering device is a device selected from a group of target rendering devices consisting essentially of a printer, a display, and a sound card.

23. The method as described in Claim 20 wherein the Web client is connected to the Web server via a non-secure connection.

24. The method as described in Claim 23 wherein the given protocol further includes the step of establishing a secure channel between the source device and a target rendering device prior to transferring the digital file.

25. The method as described in Claim 24 wherein the step of establishing a secure channel includes generating a secret key shared by the source device and the target rendering device.

26. The method as described in Claim 25 wherein the source device encrypts the digital file with the secret key as the source device transfers the digital file to the target rendering device, and wherein the target rendering device decrypts the digital file with the secret key upon receipt.

27. A computer program product in computer-readable media for use in a Web client having a source device and one or more target rendering devices, the computer program product comprising:

means for establishing a count of a number of permitted copies of a digital file located at the source device;

means, responsive to a given protocol, for transferring one or more copies of the digital file from the source device to the one or more target rendering devices;

means, responsive to each transfer, for logging an indication that the digital file has been transferred to facilitate payment of a given consideration to a provider of the digital file; and

means responsive to the logging means for adjusting the count.

28. The computer program product as described in Claim 27 further including means responsive to a given occurrence for transferring the indication to a central authority.

29. The computer program product as described in Claim 28 wherein the given occurrence is establishing a dialup connection between the Web client and an Internet Service Provider.

30. A computer system connected to a computer network and including a source device and one or more target rendering devices, comprising:

a processor;

an operating system;

an application for managing use of digital material, comprising:

means for establishing a count of a number of permitted copies of a digital file located at the source device;

means, responsive to a given protocol, for transferring one or more copies of the digital file from the source device to the one or more target rendering devices;

means, responsive to each transfer, for logging an indication that the digital file has been transferred to facilitate payment of a given consideration to a provider of the digital file; and

means responsive to the logging means for adjusting the count.

31. The computer system as described in Claim 30 wherein the application further includes means for restricting transfer of the digital file when the count reaches a given value.

32. A data processing system, comprising:

a remote control unit; and

a base unit connectable to a monitor for providing Internet access under the control of the remote control unit, the base unit comprising:

a processor having an operating system;

a browser application run by the operating system;

a secure disk storage in which a digital file is stored;

one or more target rendering devices; and

means for restricting a number of copies of the digital file that may be transferred between the secure disk storage and the one or more target rendering devices.

33. The data processing system as described in Claim 32 wherein the restricting means includes means responsive to a given occurrence for transmitting an indication of a number of copies of the digital file that were transferred between the secure disk storage and the one or more target rendering devices during a given time interval.

34. The data processing system as described in Claim 33 wherein the given occurrence is a dialup connection of the data processing system to an Internet Service Provider.

35. A management server for use in managing collection and allocation of royalties among content providers, the management server connected in a computer network to an access provider servicing a plurality of Web client appliances receiving dialup access to Web content, the management server comprising:

means for establishing an account for each of set of given content providers, each account including a representation of a given royalty value; and

means for adjusting the given royalty value in the account of the given content provider in response to receipt of an indication that a given digital file associated with the given content provider has been transferred from a source to a target rendering device in a given Web client appliance.

36. A copy management system, comprising:

a first device and a second device, each of which is certified to operate under a given security protocol;

means for establishing a secure link between the first and second devices; and

means responsive to establishment of the secure link for managing transfer of a permitted number of copies of a digital file between the first and second devices in accordance with copy control information restrictions associated with the digital file.

FIG. 1

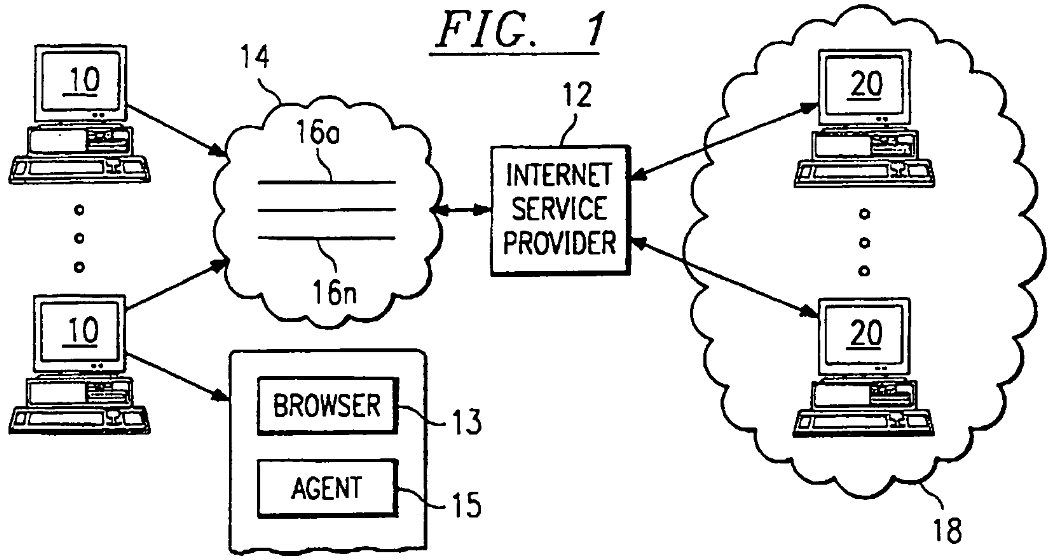


FIG. 2

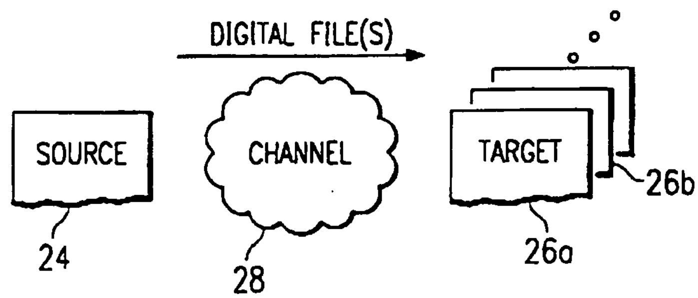


FIG. 3

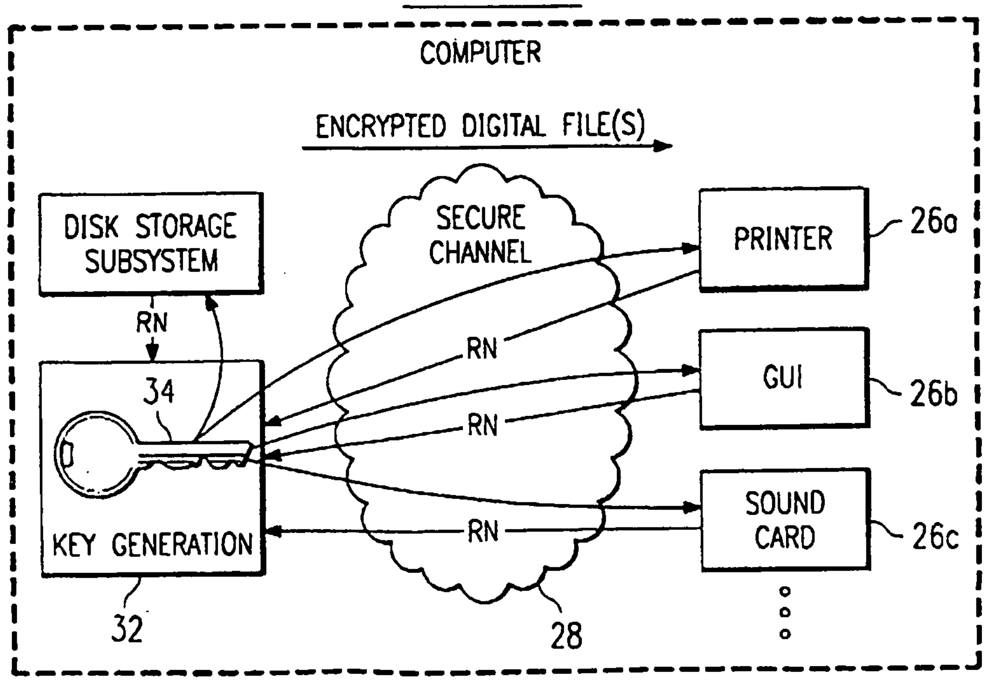
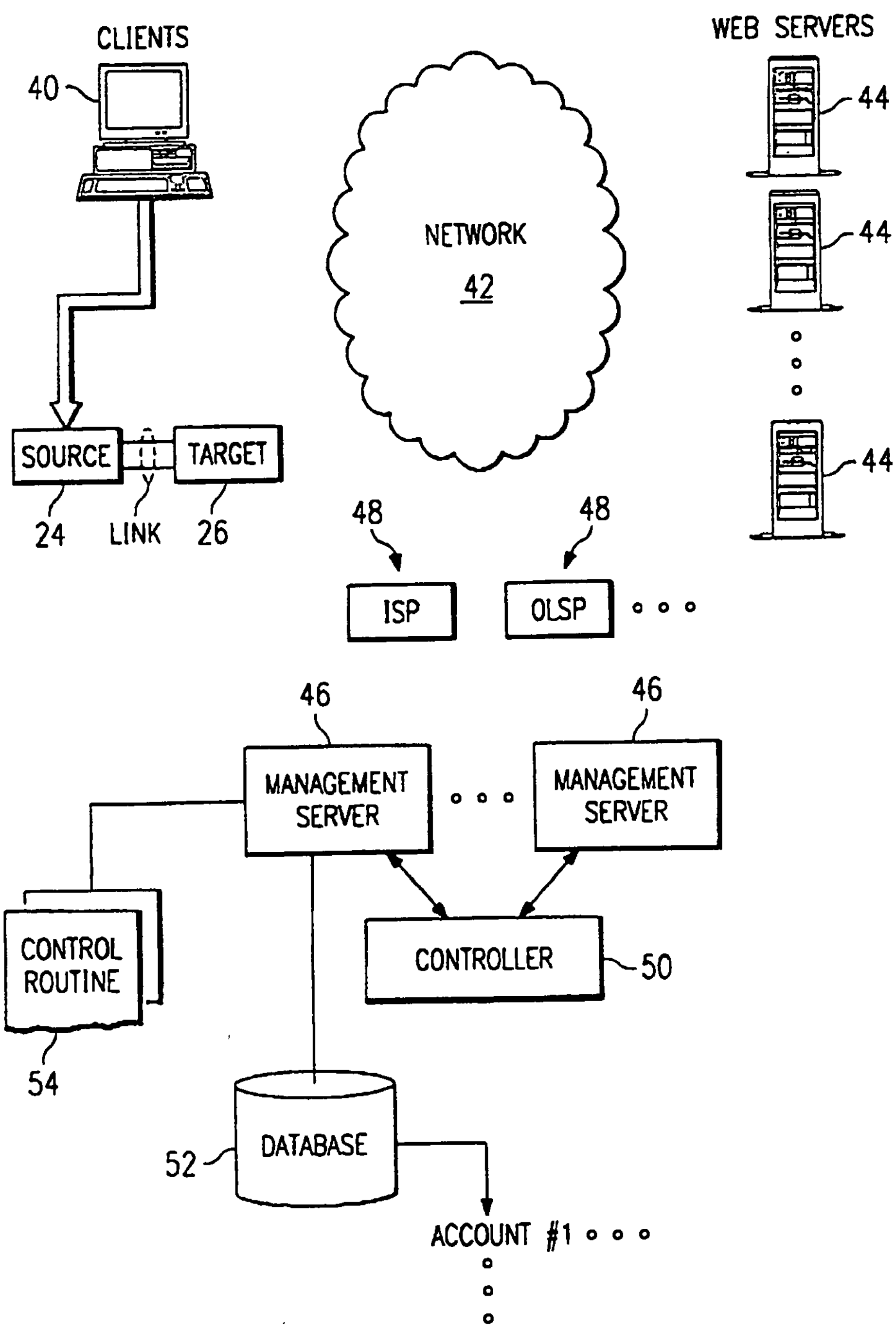


FIG. 4


CLIENTS
40
WEB SERVERS
44
44
44
NETWORK
42
SOURCE
TARGET
24
LINK
26
48
48
ISP
OLSP
46
46
MANAGEMENT SERVER
MANAGEMENT SERVER
CONTROL ROUTINE
CONTROLLER
50
54
DATABASE
52
ACCOUNT #1

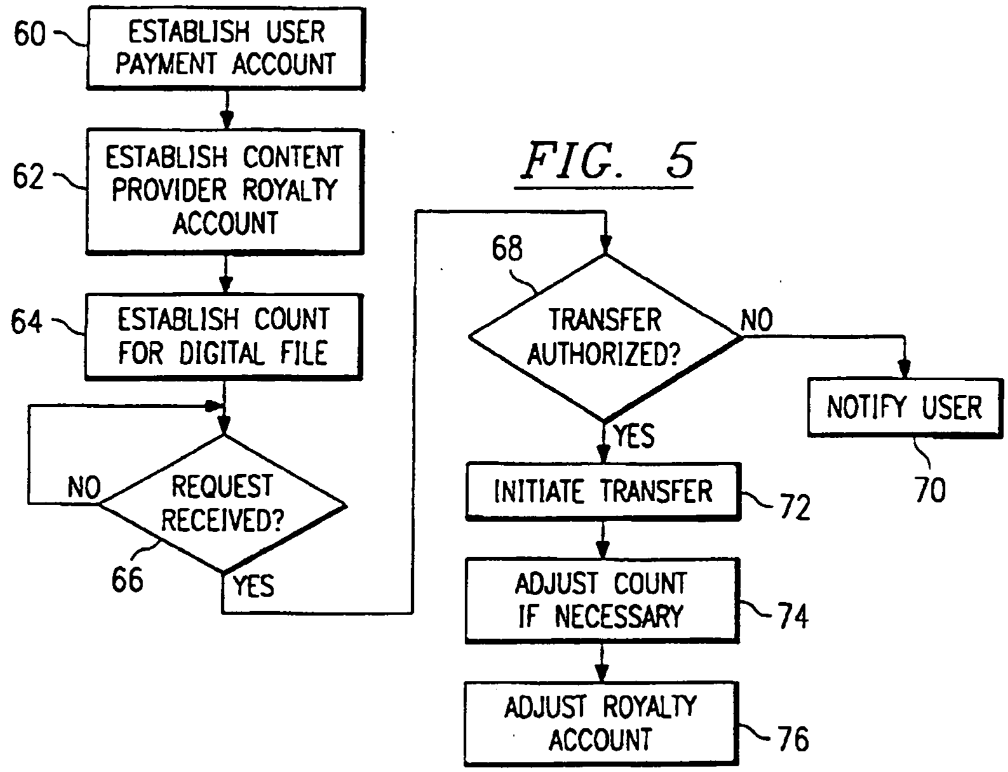

60
ESTABLISH USER PAYMENT ACCOUNT
62
ESTABLISH CONTENT PROVIDER ROYALTY ACCOUNT
64
ESTABLISH COUNT FOR DIGITAL FILE
NO
REQUEST RECEIVED?
66
YES
68
TRANSFER AUTHORIZED?
NO
NOTIFY USER
70
YES
INITIATE TRANSFER
72
ADJUST COUNT IF NECESSARY
74
ADJUST ROYALTY ACCOUNT
76

FIG. 5

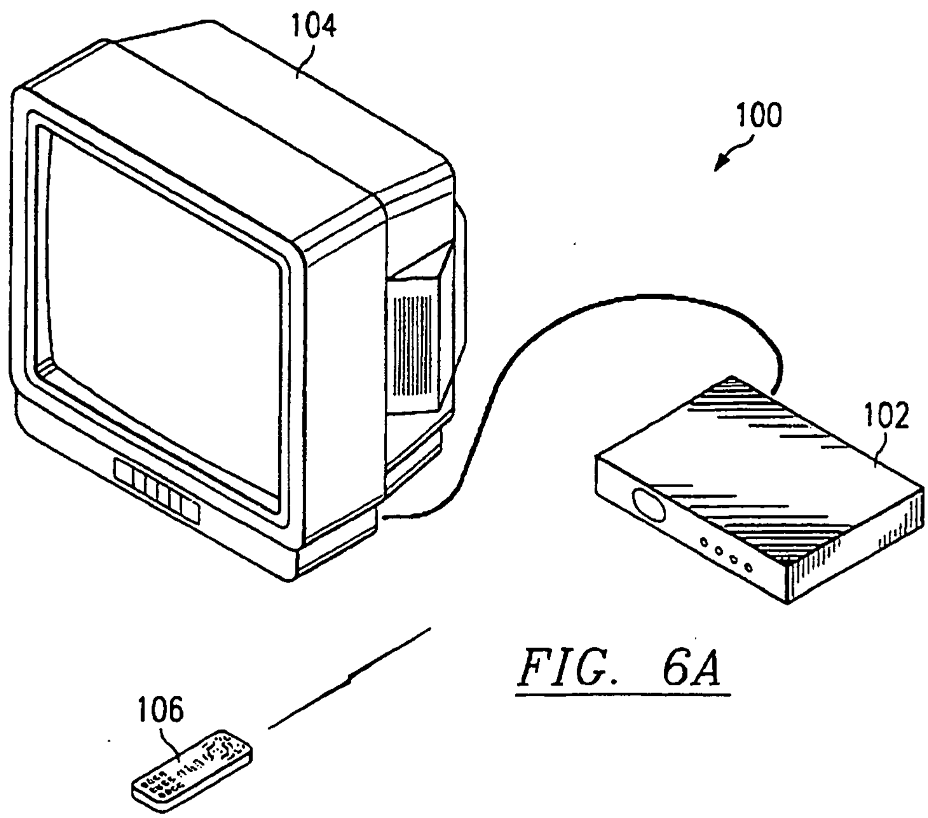

104
100
102
106

FIG. 6A

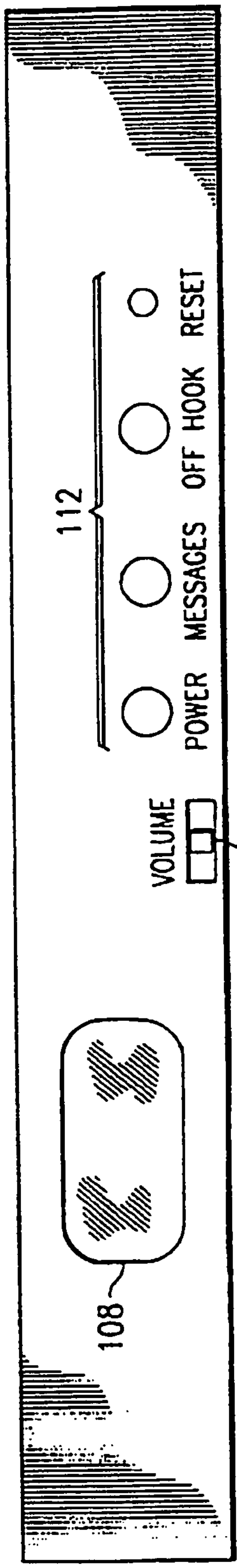

108
VOLUME
110
POWER
MESSAGES
OFF HOOK
RESET
112

FIG. 6B

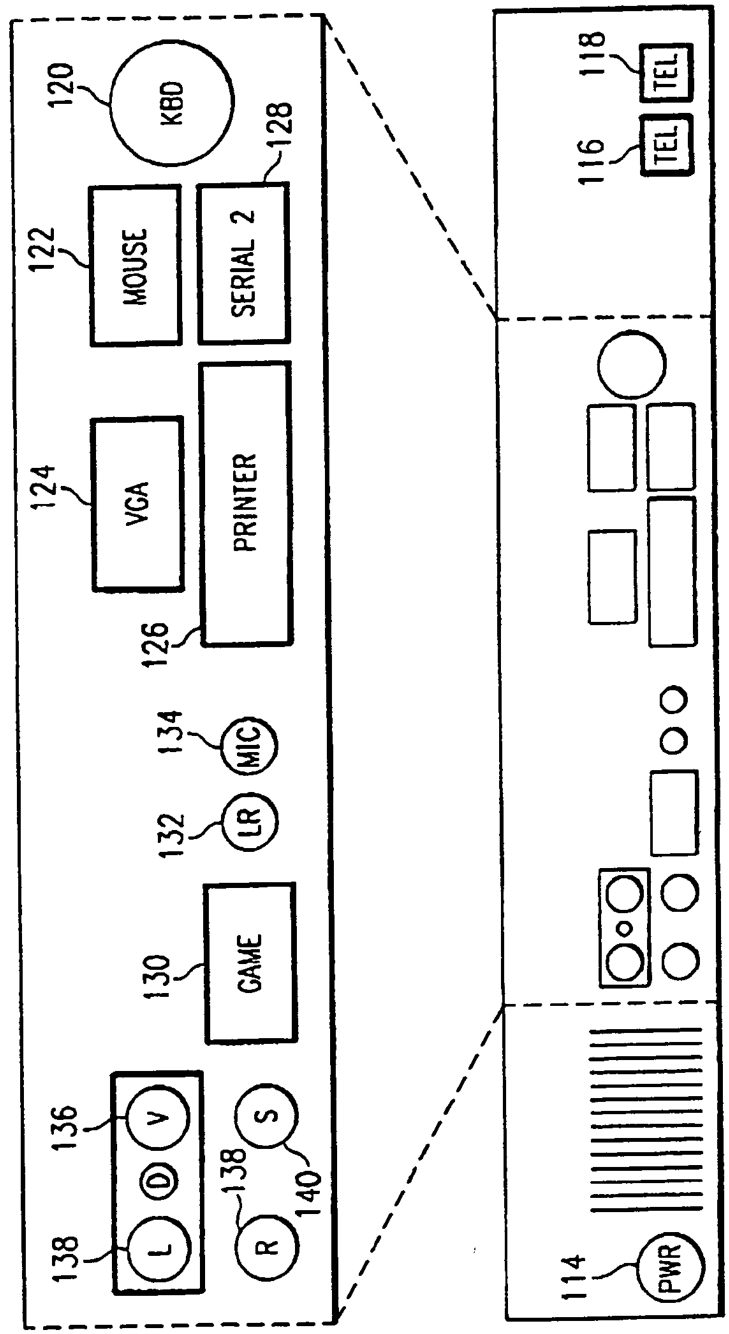

120
KBD
122
MOUSE
128
SERIAL 2
124
VGA
126
PRINTER
130
GAME
132
LR
134
MIC
136
V
D
L
138
R
S
140
114
PWR
116
TEL
118
TEL

FIG. 6C

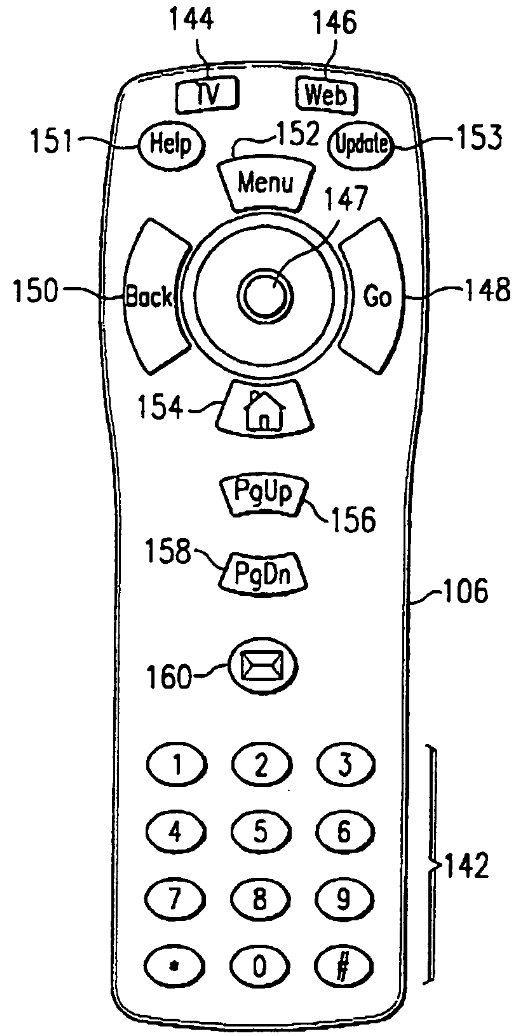
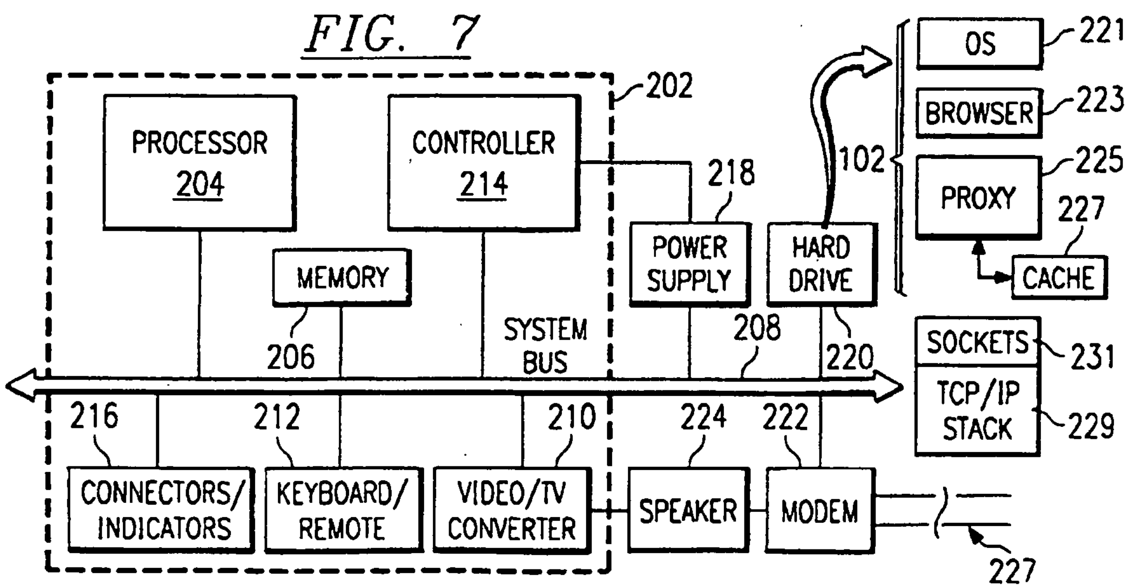

144
146
TV
Web
151
Help
152
Update
153
Menu
147
150
Back
Go
148
154
PgUp
156
158
PgDn
106
160
1 2 3
4 5 6
7 8 9
* 0 #
142

FIG. 6D

FIG. 7


202
PROCESSOR
204
CONTROLLER
214
218
POWER SUPPLY
HARD DRIVE
102
OS
221
BROWSER
223
PROXY
225
227
CACHE
MEMORY
206
SYSTEM BUS
208
220
SOCKETS
231
TCP/IP STACK
229
216
212
210
224
222
CONNECTORS/ INDICATORS
KEYBOARD/ REMOTE
VIDEO/TV CONVERTER
SPEAKER
MODEM
227

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/GB 98/03828

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F1/00

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)
IPC 6 G06F

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.
Y	US 5 532 920 A (HARTRICK THOMAS V ET AL) 2 July 1996 see figures 1,2 see column 6, line 44 - column 7, line 15 see column 14, line 51 - column 16, line 25 ---	1-20, 23-25, 27-33, 35,36
Y	EP 0 798 906 A (SUN MICROSYSTEMS INC) 1 October 1997 see figures 1-3 see column 4, line 21 - column 5, line 33 -----	1-20, 23-25, 27-33, 35,36

☐ 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	Date of mailing of the international search report
31 March 1999	08/04/1999

Name and mailing address of the ISA	Authorized officer
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	Weiss, P

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int .ional Application No
PCT/GB 98/03828

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5532920 A	02-07-1996	EP 0567800 A	03-11-1993
		JP 2659896 B	30-09-1997
		JP 6103286 A	15-04-1994
EP 0798906 A	01-10-1997	US 5761421 A	02-06-1998
		JP 10105529 A	24-04-1998

Form PCT/ISA/210 (patent family annex) (July 1992)

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/32 A2

(11) International Publication Number: WO 99/60750

(43) International Publication Date: 25 November 1999 (25.11.99)

(21) International Application Number: PCT/FI99/00432

(22) International Filing Date: 18 May 1999 (18.05.99)

(30) Priority Data:
981132 20 May 1998 (20.05.98) FI

(71) Applicant (*for all designated States except US*): NOKIA NETWORKS OY [FI/FI]; Keilalahdentie 4, FIN-02150 Espoo (FI).

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): USKELA, Sami [FI/FI]; Puistokaari 8 B 12, FIN-00200 Helsinki (FI).

(74) Agent: KOLSTER OY AB; Iso Roobertinkatu 23, P.O. Box 148, FIN-00121 Helsinki (FI).

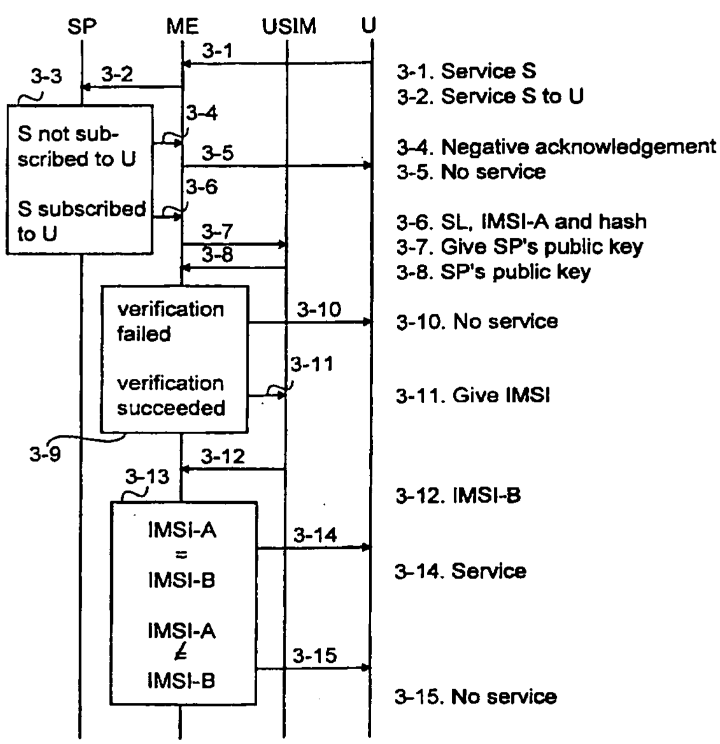
(81) Designated States: AE, AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, 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
In English translation (filed in Finnish).
Without international search report and to be republished upon receipt of that report.

(54) Title: PREVENTING UNAUTHORIZED USE OF SERVICE

(57) Abstract

A method, a system, a network element and an apparatus of a telecommunication system for preventing unauthorized use of a service. The method, in which a service request is received from a user and the service is generated by means of a service logic, is characterized in that to prevent unauthorized use of the service, authentication data is appended to the service logic (3–6), the user requesting the service is authenticated by means of the authentication data (3–9), and the service logic is executed (3–14) only if the authentication succeeds.



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		

PREVENTING UNAUTHORIZED USE OF SERVICE

BACKGROUND OF THE INVENTION

The invention relates to preventing unauthorized use of services and especially to preventing unauthorized use of the services in a mobile communication system.

Mobile communication systems were developed, because there was a need to allow people to move away from fixed telephone terminals without affecting their reachability. The services offered through mobile stations have developed along with the mobile communication systems. At the moment, various new forms of service are being planned for the current and particularly for the future third-generation mobile communication systems, such as Universal Mobile Telecommunication System (UMTS) and International Mobile Telecommunication 2000 (IMT-2000). UMTS is being standardized by ETSI (European Telecommunications Standards Institute), whereas ITU (International Telecommunications Union) is standardizing the IMT-2000 system. These future systems are very similar in basic features. The following will describe in greater detail the IMT-2000 system whose architecture is illustrated in Figure 1.

Like all mobile communication systems, IMT-2000 produces wireless data transmission services to mobile users. The system supports roaming, in other words, IMT-2000 users can be reached and they can make calls anywhere within the IMT-2000 system coverage area. IMT-2000 is expected to fulfil the need for a wide range of future services, such as virtual home environment (VHE). With the virtual home environment, an IMT-2000 user has access to the same services everywhere within the coverage area of the system. According to present knowledge, a flexible implementation of various services and especially supporting roaming requires the loading of certain service logics into the terminal of the user and/or the serving network. A serving network is the network through which the service provider offers his service to the end-user. A service logic is a program, partial program, script or applet related to the service. The service is generated by means of the service logic by executing at least the service logic and the functions defined in it. A service can also comprise several service logics.

A problem with the arrangement described above is that it does not in any way verify that the user really has the right to use the service. It is especially easy to copy and make unauthorized use of services in which the service logic is loaded into the terminal and/or serving network.

BRIEF DESCRIPTION OF THE INVENTION

Thus, it is an object of the invention to develop a method and an apparatus implementing the method so as to solve the above-mentioned problem. The object of the invention is achieved by a method, a system, a network element and an apparatus characterized by what is stated in the independent claims. The term apparatus refers here to a network element of the serving network, a terminal or any other corresponding service platform, into which the service logic can be loaded. The preferred embodiments of the invention are set forth in the dependent claims.

The invention is based on the idea of forming a service logic of two parts: user authentication and the actual service logic. The data required for user authentication is appended to the service logic, and the user is always authenticated before executing the actual service logic. This provides the advantage that an unauthorized use and copying of the service logic can be prevented. Only the users, to whom the service is subscribed and who thus have the right to use the service, can use it.

In a preferred embodiment of the invention, the service provider is always verified before the service is executed. This improves considerably the security of the user and a possible service platform into which the service logic is loaded. This ensures that the service logic truly originates from the service provider.

In a preferred embodiment of the invention, subscriber identification used to individualise a user is used in user authentication. This provides the advantage that subscriber authentication is simple, but reliable.

In a preferred embodiment of the invention, the service logic is saved with its user and authentication data in the memory of the service platform where it is loaded, and for a new user, only the authentication data of the new user is loaded. This provides the advantage that the service logic need not be loaded several times consecutively, which reduces the network load.

BRIEF DESCRIPTION OF THE DRAWINGS

In the following, the invention will be described in more detail in connection with preferred embodiments and with reference to the attached drawings in which

Figure 1 illustrates the IMT-2000 architecture,

Figure 2 shows a flow chart of the service platform functions in a first preferred embodiment of the invention,

Figure 3 is a signalling diagram of a second preferred embodiment of the invention, and

Figure 4 shows the operation of a network element controlling a service of a service provider in a third preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention can be applied to any data transmission system in which the user can receive the subscribed services in any terminal supporting service provision. In the following, the invention will be described using the IMT-2000 system as an example, without limiting the invention to this particular system, however. The specifications of mobile communication systems in general and those of the IMT-2000 and UMTS system in particular evolve rapidly. This evolution may require extra changes to the invention. Therefore, all terms and expressions should be interpreted as widely as possible and they are intended to describe and not to limit the invention. It is the function that is essential for the invention and not in which network element or apparatus it is executed.

Figure 1 shows the network architecture of the IMT-2000 system on a general level, because the system specifications are currently being defined. A more detailed network structure bears no essential significance with regard to the invention. Third-generation mobile communication systems separate a service provider SP and a network operator from each other. A service provider offers services to an end-user through a network SN of one or more network operators. This type of network SN is called a serving network. A service provider can offer services through a serving network SN of one or more network operators. In addition, a service provider may switch to another serving network during the service without the user noticing it. A service provider can also be a network operator. A serving network SN comprises an actual access network AN, one or more core networks CN, and an interworking unit adapting interfaces IWU for each different type of core network. According to present knowledge, an access network comprises base stations BS and radio network controllers RNC controlling them (not shown in the figure). A core network can be a network according to the pan-European mobile communication system GSM (Global System for Mobile Communication). Connections to other networks ON are established through a core network CN.

In the example in Figure 1, a home location register with IMT-2000 enhancement HLRi and the service control node SCN have been located in the serving network SN. The enhanced home location register HLRi contains not only the home register data of the core network but also the subscriber and service data required by the IMT-2000 system. The service provider SP maintains this IMT-2000 data for the part of the services. The subscriber makes an order agreement with the service provider which then charges the subscriber for the use of the services. The service control node SCN is a service platform to which the service logic related to the service can be loaded and in which it can be executed. The service control node SCN can also take care of loading the service elsewhere in the network and forward service requests from the user to the service provider. In addition to this, the service control node SCN makes sure that the services of the home network are also available in the visited networks.

In third-generation mobile communication networks, subscriber and user are also separated. The subscriber grants the user access to the subscribed services by giving the user an identification card (IC Card), for instance a USIM card (User and Services and Identity Module). The user accesses the services with a mobile terminal MT which is connected through base stations BS to a serving network SN over a radio path. A mobile terminal MT comprises actual mobile equipment ME and a detachably connected identification card USIM, also called a subscriber identity module. It is a smart card which can be detached from the mobile terminal and with which the subscriber can use a card-controlled mobile terminal. The user is identified by the card in the mobile terminal and not by the terminal itself. According to present knowledge, the USIM card is a multi-functional card and supports mobile communication system applications and other applications, such as Java applications, healthcare applications, etc. The subscriber can subscribe to the services of several different service providers with the same subscriber identity module USIM. The subscriber and the user can be one and the same person. The subscriber identity module USIM also contains an international mobile subscriber identity IMSI with which the subscriber can be explicitly identified and which can also be used to identify the user. The identifier of a mobile subscriber is called subscriber identity.

The terminal selection of third-generation systems will probably be extremely versatile. The terminal can be a simplified terminal for speech only or it can be a terminal providing diverse services, which acts as a service platform and supports the loading and execution of various service logics.

A mobile communication system implementing the functionality of the present invention comprises not only means required for generating and loading services according to prior art, but also means for appending authentication data to a service logic and means for authenticating the user prior to executing the service logic. Here, appending also refers to embedding data into the service logic. In addition, the system can comprise means for verifying the service provider and means for saving the service logic with its supplementary data into the memory and means for receiving plain authentication data. The means for appending authentication data and the possible means for appending verification data are preferably located together with the means required for loading the service logic of the service provider. The other means are preferably located on the service platform, for instance in the terminal or the service control point of the network operator. The means or a part of them can also be located elsewhere, for instance in the network node of the subscriber network or in the serving support node of the core network.

Figure 2 shows a flow chart of the operation according to the first preferred embodiment of the invention on a service platform which can be actual mobile equipment ME or a service control node SCN, for instance. In the first preferred embodiment of the invention, an encryption technique, known per se, based on public keys is utilized in a novel and inventive manner. One such encryption technique is RSA (Rivest Shamir Adleman public-key cryptographic algorithm) which can be used for both encryption and digital signature. In the first preferred embodiment of the invention, at least the secret key of the subscriber and the public key of the service provider are saved in the subscriber identity module USIM. If the subscriber has several key pairs, the secret key of the pair, whose public key has been entered in the subscriber data of the user, is saved. Correspondingly, a service provider can have several key pairs of which one, for instance, is saved in the subscriber identity module and information on the pair, whose key is saved in the identity module, is entered in the subscriber data. This ensures that the secret and public key of the same pair is used. In the first preferred embodiment of the invention, the service provider is verified by a digital signature. It is generated in the first embodiment of the invention by calculating a one-way hash (one-way hash function) from the service logic, which is then encrypted. This embodiment provides the unexpected advantage that in connection with the verification of the service provider, the fact whether the service logic has been changed, is also checked. If the service logic has been changed, the hash calculated from it also changes and the service provider verification does not succeed anymore.

With reference to Figure 2, a service request concerning a service S1 is received from a user U1 in step 200. In step 201, a check is made to see whether the service logic SL1 related to the service S1 is in the memory. If it is not, the service request is forwarded to the service provider in step 202. The service provider finds the actual service logic SL1 related to the service S1 and appends to the service logic authentication data A1 required for user authentication, which in the first preferred embodiment is the public key of the subscriber. After this, the service provider calculates the hash from the actual service logic and the authentication data and appends it as verification data V1 to the service logic and encrypts the thus created file with its own secret key. The file contains the verification data V1, the authentication file A1 and the actual service logic SL1. Alternatively, the service provider could encrypt the hash with its secret key, append the encrypted hash as the verification data V1 to the file and then encrypt the file with the public key of the user. After this, in step 203, the file, i.e. the actual service logic SL1 related to the service S1, the authentication data A1 appended to it for the user U1, and the verification data V1 calculated from them, is loaded onto the service platform. In step 204, the service logic SL1 is saved and in it, the authentication data A1 and the verification data V1 related to the user U1, and, of course, information on the user U1 to which the authentication data A1 and the verification data V1 are related. The data is stored in encrypted format in the memory. Then, in the first preferred embodiment, the public key of the service provider is requested from the subscriber identity module USIM in the terminal of the user in step 205 and received in step 206, after which the encryption of the service logic SL1, the authentication data A1 and the verification data V1 is decrypted in step 207 using the received key. In embodiments in which the service provider only has one key pair, the information on the public key of the provider can already be on the service platform and need not be separately requested. When the encryptions have been decrypted, the service provider is verified by calculating a hash from the service logic and the authentication data in step 208 and by comparing the thus calculated hash with the verification data V1 in step 209. If they are the same, the verification of the service provider succeeds, and after this, a challenge, i.e. a character string, is selected in step 210. How the challenge is selected bears no significance with regard to the invention. A simple and safe solution is to use a random number generator, whereby the challenge is a random number. The selected challenge is encrypted in step 211 with the public key of the subscriber, i.e. the authentication data A1. After this, in step 212, the encrypted challenge is sent to the subscriber identity module USIM in the terminal of the user U1, which decrypts the encrypted challenge into plain text with the secret key of the subscriber and sends the plain text back to the service platform. In step 213, the service platform receives the plain text and in step 214, it compares the original challenge with the plain text. If the character strings are the same, user authentication succeeds and the actual service logic SL1 can be executed in step 215.

If it is detected in step 209 that the calculated hash is not the same as the verification data, the service provider verification fails or the service logic has been changed. In both cases, executing the service logic would be a security risk and, therefore, it is not executed, and in step 217, all data saved for the service S1, i.e. the service logic SL1 and all appended authentication and verification data with user data, is deleted from the memory.

If it is detected in step 214 that the challenge is not the same as the plain text, authentication does not succeed and the service logic is not executed, and in step 216, the authentication data A1, verification data V1 and information on the user U1 appended to the service logic SL1 of the service S1, is deleted from the memory. This way, the actual service logic SL1 need not be loaded next time, only the authentication data and verification data.

If it is detected in step 201 that the service logic SL1 related to the service S1 is in the memory, a check is made in step 218 to see if authentication and verification data for the user U1 is appended to it. If this data, too, is in the memory, operation continues from step 205 where the public key of the service provider is requested from the subscriber identity module USIM. From step 205 onward, operation continues as described above. This way, network resources are saved, because the once loaded data need not be loaded again.

If it is detected in step 218 that no authentication and verification data for the user U1 is appended to the service logic SL1 in the memory, in step 219, the authentication and verification data for the service S1 is requested from the service provider for the user U1. The authentication data A1 and the verification data V1 are received in step 220, after which they and information on the user U1 are appended to the service logic SL1 in step 221. After this, operation continues from step 205 where the public key of the service provider is requested from the subscriber identity module USIM. From step 205 onward, operation continues as described above.

The service platform can be actual mobile equipment ME or a network element of the serving network, such as the service control node SCN. The memory where the data and service logics are saved can also be a cache memory. In embodiments in which the service logic is saved in the memory, the service platform can comprise means for deleting the service logic from the memory for predefined reasons, for instance after a certain time period.

In embodiment in which the service logic is not saved in the memory, steps 200, 202, 203 and 205 to 215 are executed. The data deletion described in steps 216 and 217 is not done, but the actual service logic is left unexecuted.

The steps described above in Figure 2 are not in absolute chronological order and some of the steps can be executed simultaneously or deviating from the given order. Other functions can also be executed between the steps. Some of the steps, such as the service provider verification, can also be left out. The essential thing is to authenticate the user before the actual loaded service logic is executed.

Figure 3 shows signalling according to a second preferred embodiment of the invention. In the second preferred embodiment, the subscriber identification IMSI is used as the authentication data. It is also assumed that the service logic is loaded into the actual mobile equipment ME and not saved in its memory.

With reference to Figure 3, user U sends information in message 3-1 to mobile equipment ME through the user interface requesting service S. The mobile equipment ME sends the service request through the serving network to service provider SP in message 3-2. The service request contains information on the required service S and the user U requesting the service. In step 3-3, the service provider checks if the service S is subscribed to the user U. If the service S is not subscribed to the user, the service provider sends through the serving network a negative acknowledgement to the service request in message 3-4 to the mobile equipment ME which forwards the information in message 3-5 through the user interface to the user U.

If the service S is subscribed to the user, the service provider retrieves the subscriber identification IMSI-A of the user U and the service logic SL related to the service S and calculates a hash from them. After this, the service provider encrypts the service logic and the related data (IMSI-A, hash) with the secret key of the service provider. In message 3-6, the service provider sends the service logic SL, the identification IMSI-A and the hash to the mobile equipment ME. In the second preferred embodiment, after receiving the message 3-6, the mobile equipment ME requests the public key of the service provider from the subscriber identity module USIM in message 3-7. The subscriber identity module USIM sends it to the mobile equipment in message 3-8, after which the mobile equipment ME verifies the service provider in step 3-9. The mobile equipment decrypts the encryption of the service logic SL, the subscriber identification A1 and the hash with the received public key and calculates a hash from the combination of the service logic and the subscriber identification IMSI-A. If the calculated hash is not the same as that received in message 3-6, the verification fails. In such a case, the service logic is not executed and the mobile equipment ME sends information on the verification failure through the user interface to the user U in message 3-10, saying, for instance, that the service is not available.

If the hash calculated in step 3-9 and the received hash are the same, the verification succeeds and the mobile equipment ME requests the subscriber identification IMSI of the user U from the subscriber identity module USIM in message 3-11. The subscriber identity module USIM retrieves the subscriber identification IMSI-B from its memory and sends it to the mobile equipment ME in message 3-12. In step 3-13, the mobile equipment authenticates the user by checking if the IMSI-A received from the service provider is the same as the IMSI-B received from the identity module. If the user passes the authentication in step 3-9 (i.e. IMSI-A is the same as IMSI-B), the mobile equipment ME executes the actual service logic SL and provides the service through the user interface to the user U in messages 3-14. If the values of the subscriber identifications IMSI differ from each other, authentication fails. In such a case, the mobile equipment does not execute the actual service logic, but informs the user U through the user interface in message 3-10 that the authentication failed, saying, for instance, that the service is not available.

The signalling messages described above in connection with Figure 3 are for reference only and can contain several separate messages to forward the same information. In addition, the messages can also contain other information. The messages can also be freely combined. In embodiment in which the service provider is not verified, the messages 3-7, 3-8 and 3-10 related to verification and step 3-9 are left out. Depending on the service providers, core network and mobile equipment, other network elements, to which various functionalities have been distributed, can take part in the data transmission and signalling.

Figure 4 shows a flow chart of a network element controlling a service of a service provider in a third preferred embodiment of the invention. In the third preferred embodiment, authentication and verification are only performed when a service logic is loaded into a visited (visiting) network or mobile equipment. The visited network is a network whose network element, into which the service is loaded, is a network element belonging to a provider other than the service provider. The third preferred embodiment utilizes both public key encryption and symmetrical encryption, such as DES (Data Encryption Standard). The latter encryption technique is used when the service logic is loaded into the mobile equipment. A common key is saved for it in both the subscriber identity module and the subscriber data of the user. In addition, the public key of the service provider is saved in the subscriber identity module for the service logics to be loaded into visited networks. Only encryption of the service logic with the secret key of the service provider prior to sending the service logic to the serving network or encryption with the common key prior to loading it in the mobile equipment is used as signature.

With reference to Figure 4, in step 400, a service request concerning a service S2 is received from a user U2. In step 401, a check is made to see if the user U2 subscribes to the service S2. If the user subscribes to the service S2, a check is made in step 402 to see if the service logic SL2 related to the service U2 requires loading into the mobile equipment ME of the user. If the service logic SL2 is loaded into the mobile equipment of the user, in step 403, a common key is retrieved from the subscriber data of the user U2 for encrypting the service logic SL2 in step 404. This common key is used both as the authentication data of the user and the verification data of the service provider. Nobody else should have any information on the common key in this case. The authentication and verification are performed in connection with the decryption of the service logic. The encrypted service logic SL2 is loaded into the mobile equipment ME in step 405. The user is authenticated and the service provider is verified in the mobile equipment, for instance by sending the encrypted service logic to the subscriber identity module USIM in the mobile equipment, which decrypts the service logic using the common key in its memory and sends the plain-text service logic to the mobile equipment. When the service logic has been executed, information concerning this is received in step 406, and the subscriber is charged for the use of the service in step 407.

If it is detected in step 402 that the service logic SL2 will not be loaded into the mobile equipment, a check is made in step 408 to see if the user U2 is in the home network area. If yes, the service logic SL2 is executed in step 409, after which operation continues from step 407 in which the user is charged for the use of the service.

If it is detected in step 408 that the user is not in the home network area, in the third preferred embodiment of the invention, the service logic SL2 must be loaded into the visited network. To do this, in step 410, the public key of the user U2 is retrieved from the subscriber data for appending it as authentication data to the service logic. In step 411, the public key of the user is appended to the service logic SL2, and they are encrypted using the secret key of the service provider in step 412. The encryption also acts as the verification data. If the service provider has several key pairs of public and secret keys, the secret key of the pair whose public key has been saved in the identity module of the user is used. The encrypted service logic, to which the authentication data is appended, is loaded into the visiting network in step 413. The network element of the visiting network verifies the service provider by decrypting the service logic using the public key of the service provider and authenticates the user, for instance in the manner described in connection with Figure 2, after which the service logic is executed. When the service logic has been executed, information concerning this is received in step 406, and the subscriber is charged for the use of the service in step 407.

If it is detected in step 411 that the requested service is not subscribed to the user, information is transmitted in step 414 that the service is not available to the user.

Above, in connection with Figure 4, it was assumed that the authentication and verification succeeded. If this is not the case, the service logic is not executed and the subscriber not invoiced. The steps described above in connection with Figure 4 are not in absolute chronological order and some of the steps can be executed simultaneously or deviating from the given order. Other functions can also be executed between the steps. Some of the steps can also be left out. The essential thing is that the authentication data is in some way appended to the service logic being loaded.

In the above embodiments, the actual service logic has been changed to ensure that the authentication and verification are done. This has been done by adding to the service logic a part taking care of the authentication and verification, which is always executed before the service logic. In some embodiments, the service logic can only be changed to ensure the authentication. In some embodiments, there is no need to change the service logic, and the authentication data and the possible verification data are appended to the service logic as separate data, and the service platform makes sure that the authentication and the possible verification are done. In these embodiments, pre-encrypted service logics can be used, which reduces the load of the network element, because encryption is done only once.

It has been assumed above in connection with Figures 2, 3 and 4 that the service provider appends the authentication data to the service logic before the encryption. The authentication data can also be appended to a pre-encrypted service logic. In such a case, the serving network or mobile equipment can also be adapted to append the authentication data to the service logic, for instance by means of the user data provided in the service request. It has been presented above that the user is authenticated only after the verification. However, the order bears no significance with regard to the invention. The user can be authenticated before the service provider is verified in embodiment in which the service provider is also verified. The data and/or service logic also need not be encrypted unless the encryption is used for authentication and/or verification. Other alternatives for authentication, verification and possible encryption than those described above in connection with the preferred embodiments can also be used. The preferred embodiments can also be combined. The essential thing is that the user is authenticated before executing the service logic at least when the service logic is loaded into the mobile equipment or visiting network. In embodiment in which the service logic is loaded into the mobile equipment, the encryption of the service logic with the public key of the subscriber can also be used as the authentication data. The subscriber is authenticated when the identity module USIM decrypts the encryption with the secret key of the subscriber. For security's sake, it is advantageous that USIM never sends even to the mobile equipment the secret key saved in it, and the decryption with the secret key is always performed in USIM. Other data for authentication and possible verification than used in the above examples can also be used. The requirements for the authentication data and possible verification data are adequate individualization, reliability and non-repudiation. Adequate individualization means that the data specifies the user at least by subscriber.

No hardware changes are required in the structure of the serving network. It comprises processors and memory that can be utilized in functions of the invention. All changes required for implementing the invention can instead be made as additional or updated software routines in the network elements into which the service logic is loaded. An example of such a network element is the service control node. Extra memory is also needed in the network element saving the loaded service logic with its supplementary data.

The structure of the service provider also requires no hardware changes. The service provider comprises processors and memory that can be utilized in functions of the invention. All changes required for implementing the invention can be made as additional or updated software routines to achieve the functionality of the invention. Extra memory may be needed depending on the embodiment of the invention. It is, however, limited to a small amount sufficient for saving the extra authentication data and the possible verification data.

The structure of the mobile equipment requires no hardware changes. It comprises processors and memory that can be utilized in functions of the invention. All changes required for implementing the invention can instead be made as additional or updated software routines in the mobile equipment which is adapted to function as a service platform. If the service logic is saved in the mobile equipment, extra memory is also needed.

In the subscriber identity module USIM, the extra memory possibly needed for implementing the invention is limited to a small amount sufficient for saving the extra authentication data, the possible verification data and the decryption algorithms possibly needed.

It will be understood that the above description and the figures related to it are only presented for the purpose of illustrating the present invention. The various modifications and variations of the invention will be obvious to those skilled in the art without departing from the scope or spirit of the invention disclosed in the attached claims.

CLAIMS

1. A method for preventing unauthorized use of a service in a mobile communication system, in which method

a service request is received from a user of the service, and

the service is generated by means of a service logic,

characterized in that in the method

authentication data is appended to the service logic (3-6),

the user requesting the service is authenticated by means of the authentication data (3-9), and

the service logic is executed only if the authentication succeeds (3-14).

2. A method as claimed in claim 1, **characterized** in that

verification data of the service provider is also appended to the service logic,

the service provider is verified in connection with user authentication (3-13), and

the service logic is executed only if the verification also succeeds.

3. A method as claimed in claim 2, **characterized** in that

a first hash calculated from the service logic is used as verification data of the service logic,

the service logic is loaded onto a service platform where it is executed to generate the service,

the service provider is verified on the service platform by calculating a second hash from the service logic, and

if the first and the second hash are the same, the verification succeeds,

if the first and the second hash differ, the verification fails.

4. A method as claimed in claim 2, **characterized** in that

the signature of the service provider is used as the verification data,

the service logic is signed by encrypting it with the secret key of the service provider, and

the service provider is verified by decrypting the encryption of the service logic with the public key of the service provider.

5. A method as claimed in any one of the above claims, **characterized** in that

the secret key of the subscriber is saved in the subscriber identity module (USIM) of the user of the service,

the public key of the subscriber is used as the authentication data,

a challenge encrypted with the public key of the subscriber is sent to the subscriber identity module located in the mobile equipment of the user requesting the service (207),

the challenge is decrypted into plain text with the secret key of the subscriber in the identity module,

the plain text is received from the identity module (208),

a check is made to see if the unencrypted challenge and the plain text correspond to each other (209), and

if they correspond, the authentication succeeds, and

if they do not correspond, the authentication fails.

6. A method as claimed in claims 1, 2, 3 or 4, **characterized** in that

individual identity of the subscriber is used as the authentication data,

a service request is received from the user (3-1),

the individual subscriber identity related to the user is requested (3-7),

the requested identity is received (3-8),

a check is made to see if the authentication data and the requested identity correspond to each other (3-9), and

if they correspond, the authentication succeeds, and

if they do not correspond, the authentication fails.

7. A method as claimed in any one of the above claims, **characterized** in that

the service logic is loaded onto the service platform where it is executed to generate the service, and

the authentication data is appended to the service logic in connection with the loading.

8. A method as claimed in claim 7, **characterized** in that

the service logic, the authentication data appended to it, and the data indicating the user are saved on the service platform in connection with the loading (204),

a service request is received from the user, a check is made to see if the service logic related to the requested service is saved on the service platform (201), and

if not, the service logic is loaded (203),

if yes,

- a check is made to see if authentication data has been saved for the user requesting the service (217), and

- if yes, the user is authenticated,

- if not,

-- authentication data is requested for the user (218),

-- the authentication data and the data indicating the user are saved in the service logic (220), and

-- the user is authenticated.

9. A telecommunication system comprising

a first part (SP) to produce the service for the user by means of a service logic, and

a second part to provide the service (SN, MT) to the user of the service,

in which system the first part (SP) is adapted to identify the user requesting the service and to check, if the service is subscribed to the user, and if the service is subscribed to the user, to generate the service by loading the service logic into the second part (SN, MT) which is adapted to provide the service by executing the loaded service logic,

characterized in that

the first part (SP) is adapted to append authentication data into the service logic being loaded for user authentication, and

the second part (SN, MT) is adapted to authenticate the user and to execute the service logic only in response to a successful authentication.

10. A system as claimed in claim 9, **characterized** in that

the first part (SP) is adapted to sign the service logic by encrypting it with an encryption key agreed with the second part, and

the second part (SN, MT) is adapted to verify the first part by decrypting the encryption of the service logic with a key corresponding to the agreed key and to execute the service logic only if the verification also succeeds.

11. A system as claimed in claim 9 or 10, **characterized** in that

the telecommunication system is a mobile communication system (IMT-2000) comprising at least one service provider and serving network,

the first part is the service provider (SP), and

the second part is the serving network (SN) comprising at least one network element (SCN).

12. A system as claimed in claim 9 or 10, **characterized** in that

the telecommunication system is a mobile communication system (IMT-2000) comprising at least one service provider (SP) and mobile terminal (MT) which is connected to the service provider through a serving network (SN) and which mobile terminal (MT) comprises in addition to actual mobile equipment (ME) a subscriber identity module (USIM) which is detachably connected to the mobile equipment,

the first part is the service provider (SP), and

the second part is the actual mobile equipment (ME).

13. A network element (SP) generating a telecommunication system service for a user, which produces the service by means of a service logic and which comprises means for identifying the user requesting the service and for checking if the service is subscribed to the user and for loading the service logic into the telecommunication system if the service is subscribed to the user,

characterized in that the network element (SP) comprises means for appending the authentication data to the service logic being loaded so that the user of the service is authenticated before the service logic is executed.

14. A network element (SP) as claimed in claim 13, **characterized** in that it comprises means for signing the service logic before it is loaded into the network.

15. A network element (SP) as claimed in claim 13 or 14, **characterized** in that it comprises a processor arranged to execute software routines, and said means have been implemented as software routines.

16. An apparatus of a telecommunication system, which apparatus comprises service logic executing means for providing a service from a service provider of a telecommunication system to a user of the service,

characterized in that the apparatus (SCN, ME) comprises separation means for separating the authentication data of a user from a loaded service logic,

authentication means responsive to the separation means for user authentication, and

service logic execution means are adapted to be responsive to the authentication means.

17. An apparatus (SCN, ME) as claimed in claim 16, **characterized** in that

it comprises verification means for service provider verification by means of verification data in the loaded service logic, and

the service logic verification means are adapted to be responsive to the authentication means.

18. An apparatus (SCN, ME) as claimed in claim 16 or 17, **characterized** in that it comprises a processor arranged to execute software routines, and said means are implemented as software routines.

19. An apparatus as claimed in claim 16, 17 or 18, **characterized** in that it is a network element (SCN) of a mobile communication system, which is adapted to function as a service platform.

20. An apparatus as claimed in claim 16, 17 or 18, **characterized** in that it is the mobile equipment (ME) in a mobile communication system.

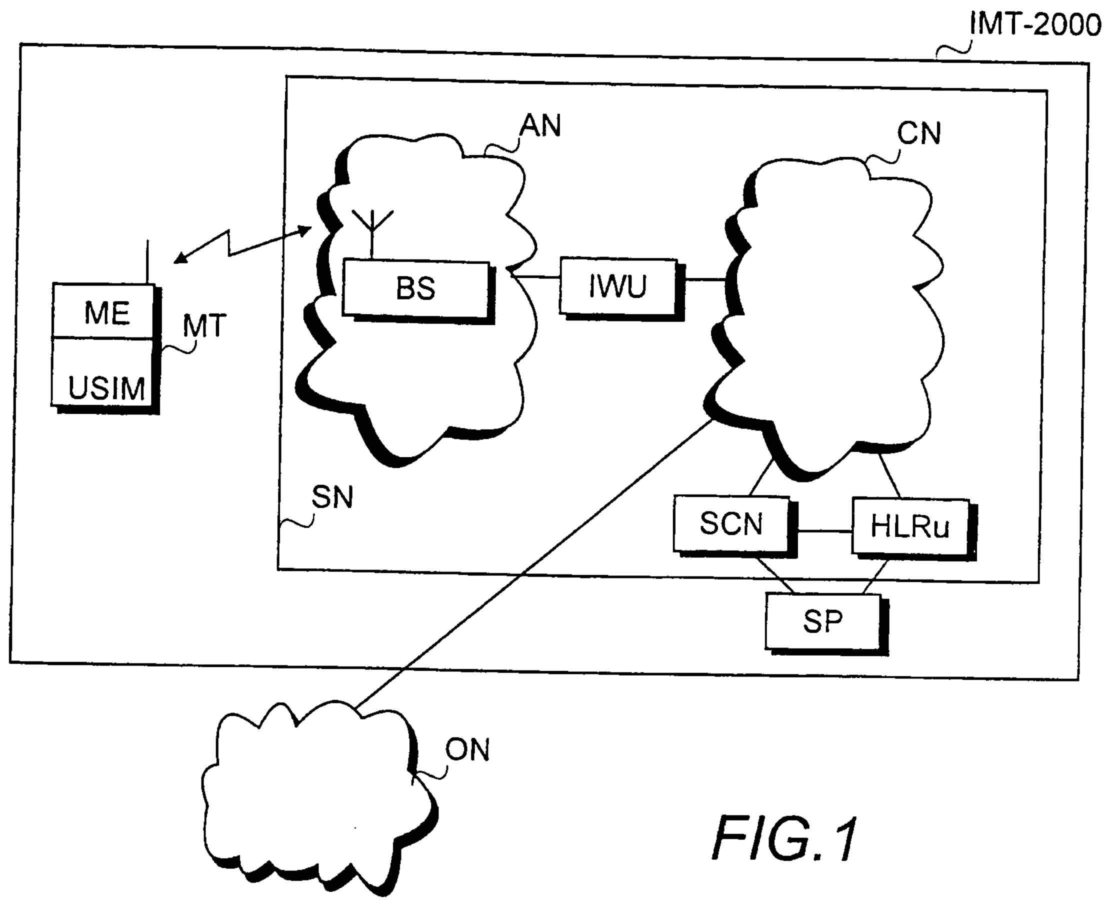
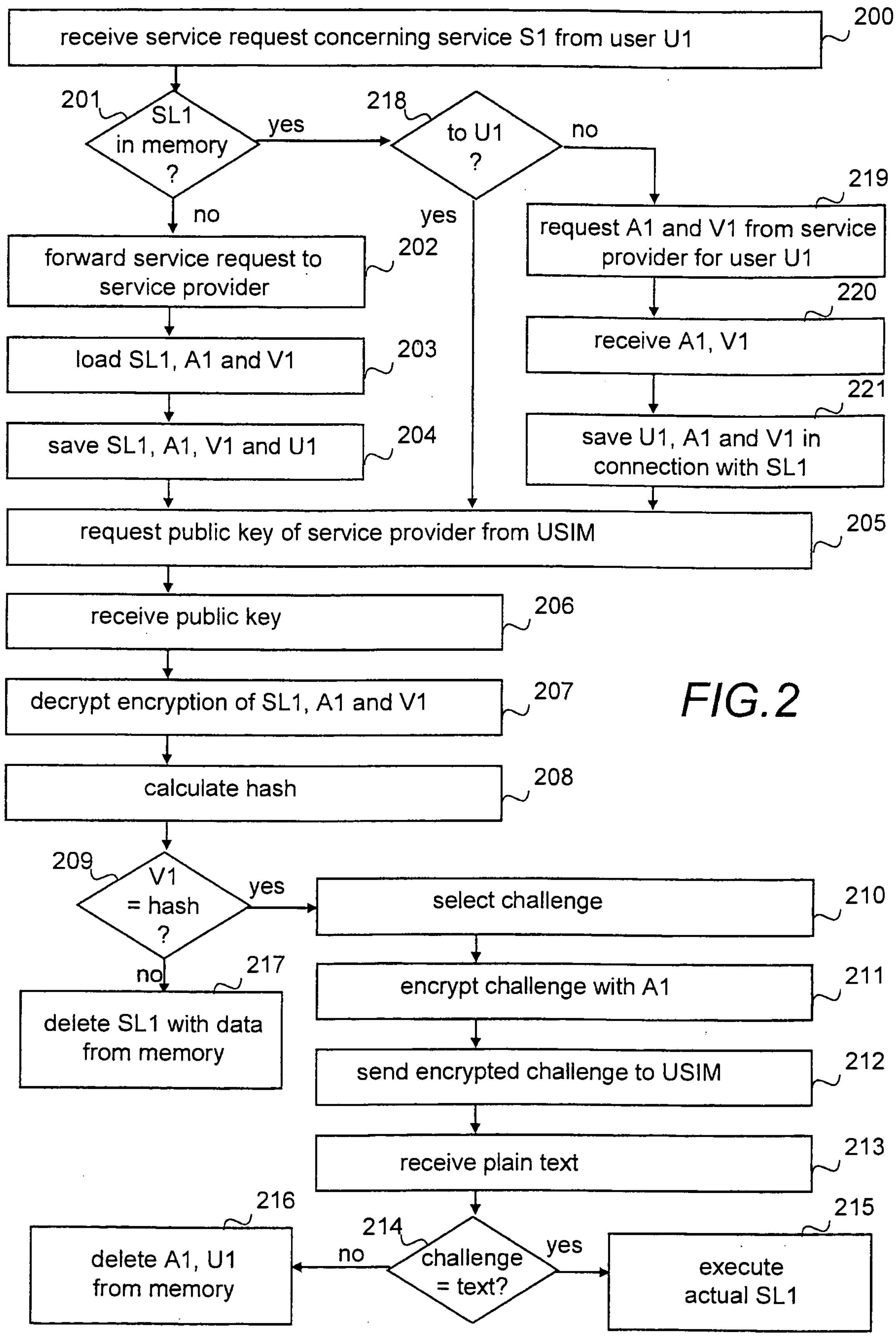


FIG.1



receive service request concerning service S1 from user U1
200
201
SL1 in memory ?
yes
no
218
to U1 ?
no
yes
219
request A1 and V1 from service provider for user U1
220
receive A1, V1
221
save U1, A1 and V1 in connection with SL1
forward service request to service provider
202
load SL1, A1 and V1
203
save SL1, A1, V1 and U1
204
request public key of service provider from USIM
205
receive public key
206
decrypt encryption of SL1, A1 and V1
207
calculate hash
208
209
V1 = hash ?
yes
no
select challenge
210
encrypt challenge with A1
211
send encrypted challenge to USIM
212
receive plain text
213
217
delete SL1 with data from memory
216
delete A1, U1 from memory
214
challenge = text?
no
yes
215
execute actual SL1

FIG.2

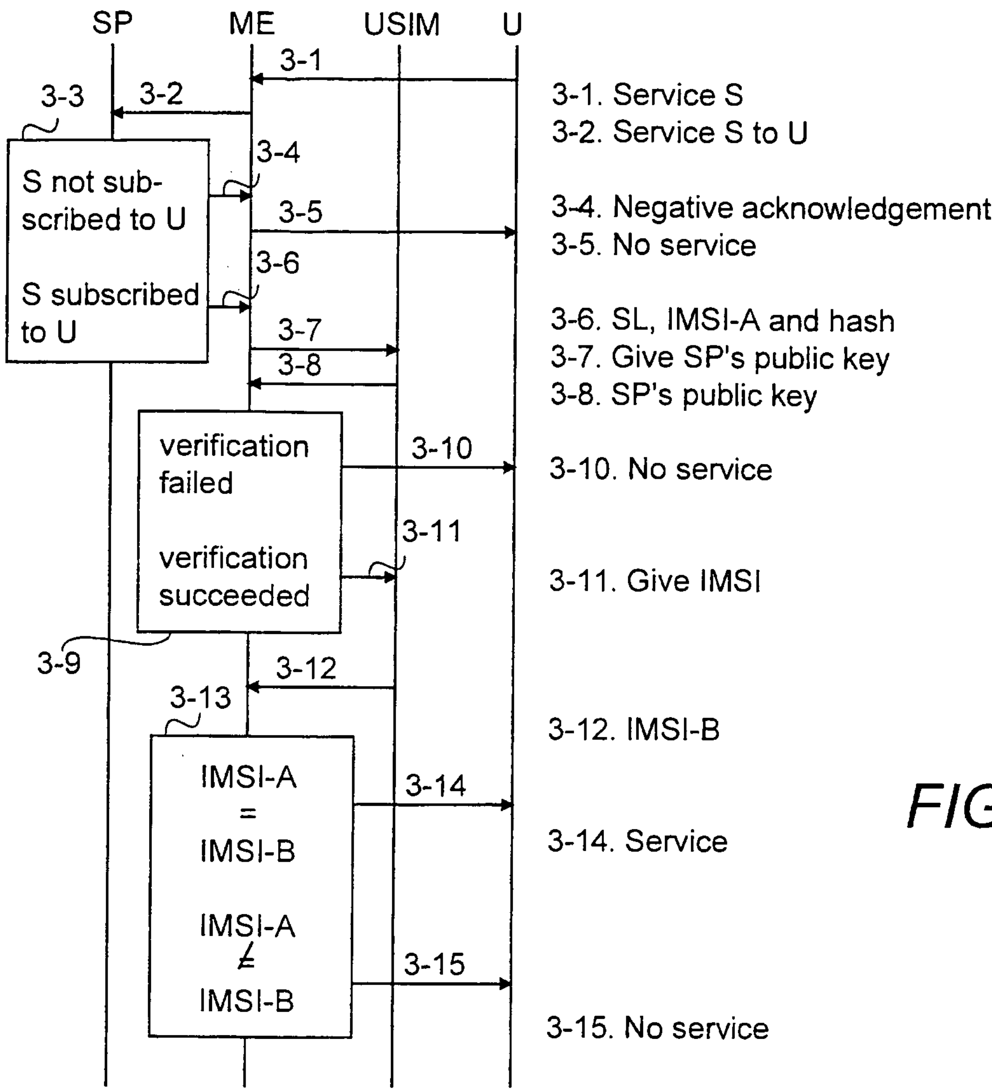

SP
ME
USIM
U
3-1
3-2
3-3
3-4
3-5
3-6
3-7
3-8
3-9
3-10
3-11
3-12
3-13
3-14
3-15
S not sub-scribed to U
S subscribed to U
verification failed
verification succeeded
IMSI-A = IMSI-B
IMSI-A ≠ IMSI-B
3-1. Service S
3-2. Service S to U
3-4. Negative acknowledgement
3-5. No service
3-6. SL, IMSI-A and hash
3-7. Give SP's public key
3-8. SP's public key
3-10. No service
3-11. Give IMSI
3-12. IMSI-B
3-14. Service
3-15. No service

FIG.3

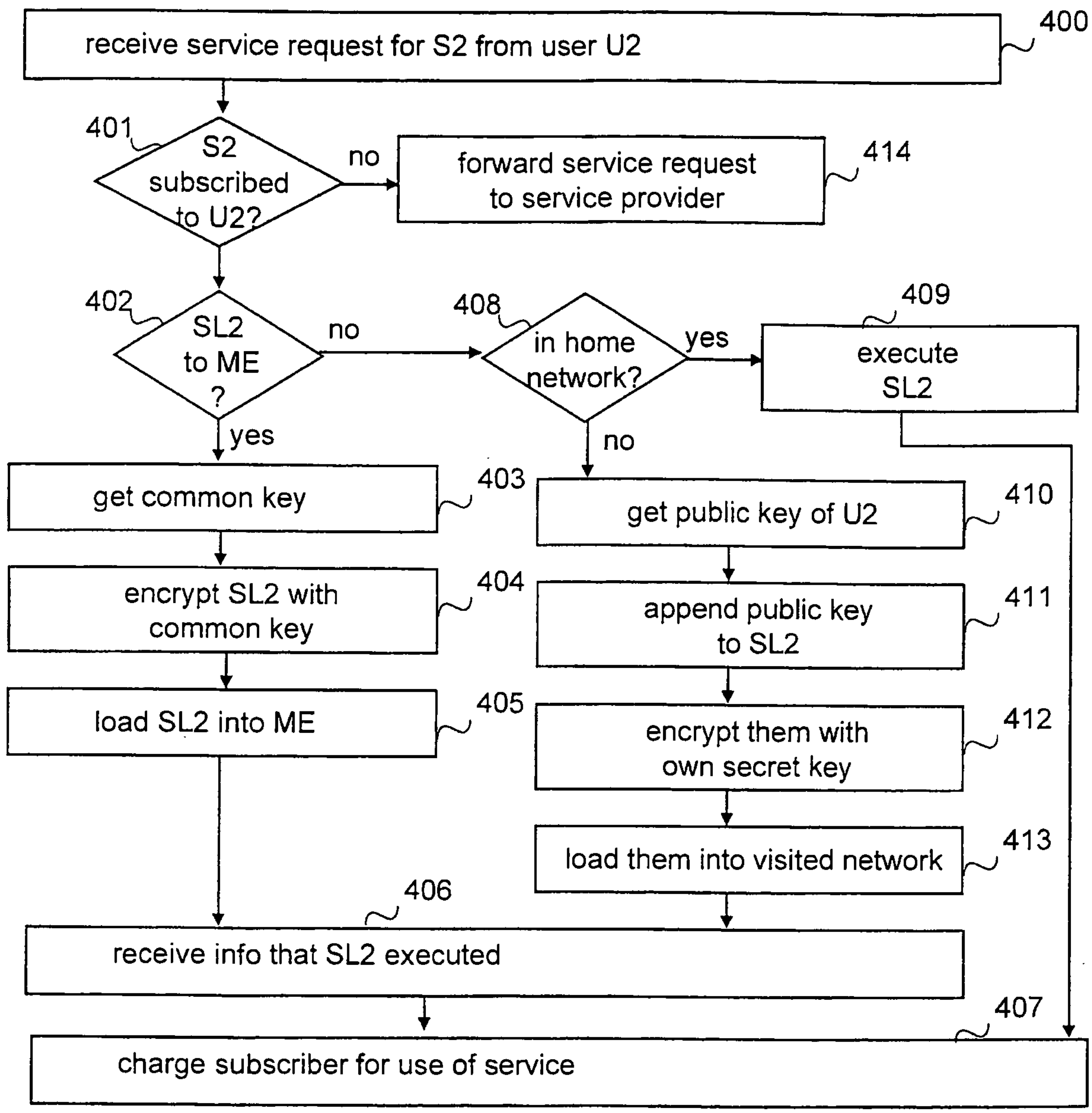

receive service request for S2 from user U2
400
401
S2 subscribed to U2?
no
forward service request to service provider
414
402
SL2 to ME ?
no
408
in home network?
yes
409
execute SL2
yes
no
get common key
403
get public key of U2
410
encrypt SL2 with common key
404
append public key to SL2
411
load SL2 into ME
405
encrypt them with own secret key
412
load them into visited network
413
406
receive info that SL2 executed
407
charge subscriber for use of service

FIG.4

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04N 7 /24

A2

(11) International Publication Number: WO 00/04727

(43) International Publication Date: 27 January 2000 (27.01.00)

(21) International Application Number: PCT/EP99/04704

(22) International Filing Date: 2 July 1999 (02.07.99)

(30) Priority Data:

60/092,726	14 July 1998 (14.07.98)	US
09/276,437	25 March 1999 (25.03.99)	US

(71) Applicant: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(72) Inventor: EPSTEIN, Michael, A.; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

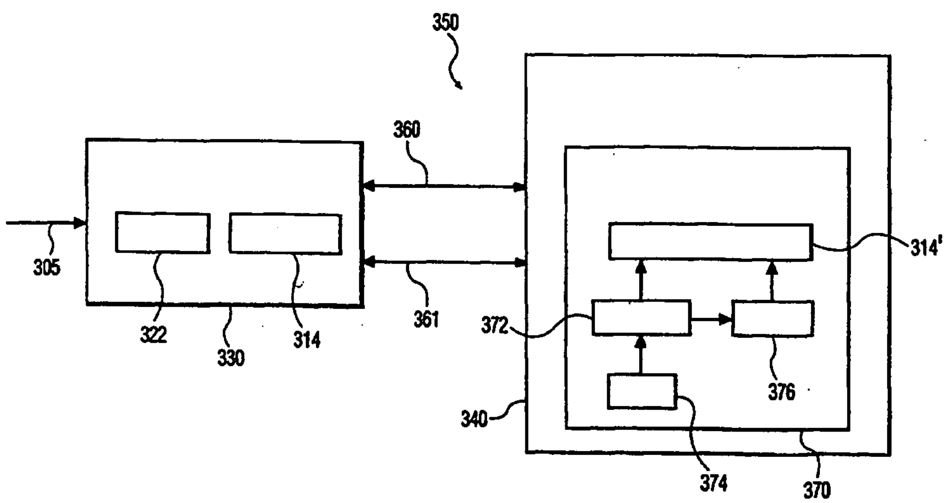
(74) Agent: GROENENDAAL, Antonius, W., M.; Internationaal Octrooibureau B.V., Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(81) Designated States: BR, CN, JP, KR, MX, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published

Without international search report and to be republished upon receipt of that report.

(54) Title: USE OF A WATERMARK FOR THE PURPOSE OF COPY PROTECTION



(57) Abstract

A copyright protection system for protecting content wherein a time dependent ticket is calculated (314) at a source device (330) by combining a checkpoint with a ticket. The checkpoint is transmitted (361) from a display device (340) to the source device prior to the source device transmitting (360) watermarked content to the display device. The checkpoint is also stored (376) at the display device. Thereafter, the source device transmits, to the display device, watermarked content, the ticket, and the time dependent ticket. At the display device, the stored checkpoint is compared (314) to a current count of a local clock (374) that was utilized for producing the checkpoint. If the stored checkpoint is within a window of time of the local clock, then the stored checkpoint is combined (314') with the ticket in the same way that the checkpoint is combined with the ticket at the source device. A result of the combination is compared to the time dependent ticket and if the result equals the time dependent ticket, then the watermark and ticket may be compared in the usual way to determine the copy protection status of the copy protected content (314').

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
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

Use of a Watermark for the Purpose of Copy Protection.

Field of the Invention

This invention generally relates to a system for protecting copyrighted content. Specifically, the present invention pertains to utilizing a ticket and a watermark to protect content.

Background of the Invention

The ability to transmit digital information securely is increasingly important. Owners of content want to be able to provide the content to authorized users without having the content utilized by unauthorized users. However, one problem with digital content is that an exact copy can be made without any degradation in the quality of the copy. Therefore, the copying of digital content is very attractive to pirating operations or attackers.

Both small-scale and commercial pirates are interested in defeating copy-protected content in order to produce and sell illegal copies of the content. By avoiding payments to the rightful owner of the copy-protected content, the pirates may reap large profits. Typically, the pirate may take advantage of the difference in release windows in order access high value content and distribute it.

For instance, in the movie industry, release windows are utilized to maximize profit from content. The essence of these release windows is to first release the content to a premium service such as a pay-per-view service or a video on demand service. Thereafter, the content may be released on a lower price service such as a home-box-office service. At this time, the content may also be available to a consumer through a purchased storage medium such as a Digital Video Disc (DVD).

Pirates however, frustrate the use of these release windows by pirating the content that is available through the premium service and then releasing pirated versions of the content to the public. This may cause substantial financial losses to the rightful owners of the content. Accordingly, a successful copy protection scheme should at least frustrate a pirates attempt for a sufficient period of time till the legitimate owner of the content may reap their rightful profits.

Beyond some level of attacker, the expense of defeating the attacker exceeds a reasonable limit whereby the device must be priced beyond what consumer is willing to pay. Thus, a copy protection solution must be cost effective but secure against a large number of attackers.

A cost-effective method of copy protection is discussed in detail by Jean-Paul Linnartz et al., in Philips Electronics Response to Call for Proposals Issued by the Data Hiding Subgroup Copy Protection Technical Working Group, July 1997 ("Linnartz"). Within a digital transmission, such as an MPEG transport stream, additional data may be embedded within the transport stream to set the copy protection status of content contained within the digital transmission. For instance, the desired copy protection status may be "copy-once", "no-more-copy", "copy-never", and "copy-freely". Content that has a status of copy-once may be played and copied. During copying, the copy-once content is altered such that the content is in the no-more-copy state. Copy-never content is content that may only be played and may not be copied. Copy-freely content may be played and copied without restriction.

The additional data may take the form of a digital watermark. The watermark may be embedded directly into the content so that removal of the watermark will degrade the quality of the content. The watermark may be utilized as part of the copy protection scheme. As an example, the copy-freely state may be designated by the lack of a watermark within the content.

In operation, a transmission, such as a digital transmission, is sent from a source device and received by a receiving device. A source device is a device that is writing content onto a data bus, initiating a broadcast transmission, initiating a terrestrial transmission, etc. A sink device is a device that reads content from the data bus, etc.

Fig. 1 shows a typical system for the transmission of content. In Fig. 1, the source device is a broadcast initiator 101 that utilizes a transmitting antenna 102 to transmit content. The sink device is a broadcast receiver, such as a set-top-box (STB) 104 that utilizes a receiving antenna 103 for receiving the transmitted content. The STB 104 is shown connected to a display device 105, a player 106, and a player/recorder 107, through a bus 108. The term bus is utilized herein to refer to any system for connecting one device to another device. The bus may be a hard wired system such as a coaxial wire, an IEEE 1553 bus, etc., or the bus may be a wireless system such as an infra-red (IR) or radio frequency (RF) broadcast system. Several of the devices shown in Fig. 1 may at one time act as a source device and at another time act as a sink device. The STB 104 may be a sink for the broadcast transmission and be a source for a transmission on the bus 108. The player/recorder 107 may be a source/sink of a transmission to/from, respectively, the bus 108.

In the copy protection scheme discussed by Linnartz, a watermark (W) is embedded within transmitted content. A ticket is transmitted along with the transmitted content. The embedded watermark and the ticket together are utilized to determine the copy protection status of the transmitted content. The watermark may be embedded into the content by at least two known methods. One method embeds the watermark (W) in the MPEG coding of the content. Another method embeds the watermark (W) in the pixel data of the content. The ticket (T) is mathematically related to the watermark (W) as discussed in more detail below.

Performing one or more one-way functions on the ticket (T) derives the watermark (W). By use of the term one-way function, what is meant is that it is computationally unfeasible to compute the inverse of the function. An example of a publicly known mathematical one-way function is a hashing function, such as secure hash algorithm one (SHA-1) or RACE Integrity Primitives Evaluation Message Digest (RIPEMD). Computing an inverse means finding which particular x_0 leads to a given y_0 with $y_0=F(x_0)$. The term unfeasible is intended to mean that the best method will take too long to be useful for a pirate. For instance, the time that is required for a pirate to compute the inverse of a hashing function is too long for the pirate to frustrate the intended release window for protected content. The most efficient method known to find such an x_0 may be to exhaustively search all possible bit combinations of x_0 and to compute and verify $F(x_0)$ for each attempt. In other cases, there may be a more efficient method than an exhaustive search to compute an inverse of a one-way function, yet these methods are still too time consuming to be feasible for the pirate.

The bit content of the ticket (T) is generated from a seed (U). The content owner provides the seed (U). From the seed (U), a physical mark (P) is created. The physical mark (P) may be embedded on a storage medium such as a Read-Only Memory (ROM) disk. Performing one or more one-way functions on the physical mark (P), produces the ticket (T). The number of functions performed on the physical mark (P) to create the ticket (T) depends on the copy protection intended for the content.

In accordance with the system, the ticket (T) changes state during every passage of a playback device (e.g., a source device) and a recording device (e.g., a sink device). As discussed above, the state modifications are mathematically irreversible and reduce the remaining copy and play rights of the content that are granted by the ticket (T). In this way, the ticket (T) indicates the number of sequential playback and recordings that may still be performed and acts as a cryptographic counter that can be decremented but not incremented.

It should be noted that the copy protection scheme only protects content on compliant systems. A compliant system is any system that obeys the copy protection rules described above and hereinafter. A non-compliant system may be able to play and copy material irrespective of the copy protection rules. However, a compliant system should refuse to play copies of content illegally made on a non-compliant system.

In accordance with the copy protection scheme, a physical mark (P) (e.g., data) is embedded on a storage medium and is not accessible by other user equipment. The physical mark (P) data is generated at the time of manufacturing of the storage medium as described above and is attached to the storage medium in a way in which it is difficult to remove the physical mark (P) data without destroying the storage medium. The application of a one-way mathematical function, such as a hashing function, to the physical mark (P) data four times results in a watermark. Much like watermarks embedded in paper, the watermark is embedded in the medium (e.g., containing video, audio, or data) in such a way that it is infeasible to remove the watermark without destroying the material. At the same time the watermark should be imperceptible when the medium is used in the usual manner, such as when content from the medium is displayed.

A watermark by itself may indicate whether or not content stored on the storage medium is copy-once or copy-never. For instance, the absence of a watermark may indicate that the content may be copied freely. The presence of the watermark without a ticket on a storage medium may indicate copy-never content.

When the content is transmitted over a bus or other transmission medium, the physical mark (P) data is hashed twice to generate a ticket. When a compliant player receives the content, the ticket is hashed twice and matched to the watermark. In the case where the twice-hashed ticket and the watermark match, the content is played. In this way, a party may not substitute a false ticket along with the content to frustrate the copy protection scheme. In the case were there is a watermark but no ticket in the content, a compliant system will refuse to record the content.

When a compliant recorder reads the content, the watermark is checked to see if the material is copy-freely, copy-once, or copy-never. When there is no watermark, the content is copy-freely and may be copied freely as discussed above. When the content contains a watermark but no ticket, the content is copy-never and a compliant recorder will refuse to copy the content. However, a compliant player will play the content as long as the ticket hashed two times matches the watermark. When the content is copy-once, the content contains both a watermark and a ticket, a compliant recorder will hash the ticket twice and compare the twice-hashed ticket to the watermark. In the case where the watermark matches the twice-hashed ticket, the content may be recorded along with a once-hashed ticket and the watermark, thereby creating copy-no-more content (e.g., content with a once-hashed ticket and a watermark). The physical mark will be different on a writable disc and thus, even if an illegal copy is made of copy-never content via a non-compliant recording device, a compliant player will refuse to play the content recorded on the writable disc.

It should be noted that in a broadcast system, such as a pay-per-view system, a copy-never state may be indicated by the presence of a once-hashed ticket and a watermark. Both copy-no-more stored content and copy-never broadcast content are treated by a compliant system similarly. The content containing the once-hashed ticket may be played but may not be recorded in a compliant system. In the event that a party tries to record the content with the once-hashed ticket, a compliant recorder will first twice-hash the once-hashed ticket and compare the result (e.g., a thrice-hashed ticket) with the watermark. Since the thrice-hashed ticket will not match the watermark, the compliant recorder will refuse to record the content.

A compliant player that receives the once-hashed ticket will hash the once-hashed ticket and compare the result (e.g., a twice-hashed ticket) to the watermark. Since the twice-hashed ticket matches the watermark, the compliant player will play the content.

However, a problem exists wherein a non-compliant recorder receives content containing a ticket (a twice-hashed physical mark) and a watermark. In the event that a non-compliant recorder does not alter the ticket upon receipt or recording (e.g., the non-compliant recorder makes a bit-for-bit copy), the non-compliant recorder may make multiple copies of the ticket and the watermark that will play on a compliant player and that may be recorded on a compliant recorder. The same problem can exist where a non-compliant recorder receives content containing a once-hashed ticket (a thrice-hashed physical mark) and a watermark indicating copy-no-more content. In this case, the non-compliant recorder may make multiple copies of the once-hashed ticket and the watermark that will play on the compliant player.

In a case wherein the player receives the content directly from a read only medium, such as a Compact Disc ROM (CD-ROM), a physical mark can be embedded in the physical medium of the CD-ROM that is produced by an authorized manufacturer. The player may then check the physical mark to ensure that the content is being received from an authorized medium. In this way, if a pirate makes an unauthorized copy, the physical mark will not be present on the unauthorized copy and a compliant player will refuse to play the content. However, in the case of broadcast data for instance, wherein a player does not read content directly from the read-only medium, this method of copy protection is unavailable. Thus, for instance, a non-compliant player may deceive a compliant display device.

Accordingly, it is an object of the present invention to overcome the disadvantages of the prior art.

Summary of the Invention

This object of the present invention is achieved by a copy protection system for protecting content, such as content containing a watermark embedded therein (e.g., watermarked content). To this end, the invention provides a content protecting method, a copy protection system, a source device, and a display device as defined in the independent claims. The dependent claims define advantageous embodiments. In accordance with the present invention, a relative time dependent ticket is created at a source device preferably utilizing a display device dependent time reference (a checkpoint). In accordance with one embodiment of the present invention, the checkpoint is combined with a ticket utilizing a concatenation function and a one-way function (e.g., a hashing function). The checkpoint is transmitted from the display device to the source device prior to the source device transmitting watermarked content to the display device. The checkpoint is also stored at the display device. Thereafter, the source device transmits to the display device watermarked content, the ticket, and the relative time dependent ticket.

At the display device, the stored checkpoint is compared to a current relative time reference. If the difference between the stored checkpoint and the current relative time reference is acceptable, then further steps, as discussed below, may proceed. What is an acceptable difference between the stored checkpoint and the current relative time reference will depend on the nature of the desired content protection. For example, in one embodiment or for one particular type of content, the difference may be short to ensure that the content is being transmitted and received in real time. In another embodiment or for another type of content, the difference may be longer to allow for storage of the content for later playback.

When the difference between the stored checkpoint and the current relative time reference is acceptable, the ticket is next hashed twice and compared to the watermark in the usual way. In the event that the ticket compares to the watermark ($W = H(H(T))$), the stored checkpoint is combined with the ticket in the same way that the checkpoint was combined with the ticket at the source device. A result of the combination is compared to the relative time dependent ticket. If the result equals the relative time dependent ticket, then the display device is provided with access (e.g., enabled to display) to the watermarked content.

Preferably, the checkpoint is derived from a counter that purposely is inaccurate such that the count can be said to be unique as compared to the count from other display devices. The counter is constructed with a sufficient number of bits such that the counter will not roll over to zero in the lifetime of the display device. The counter is constructed to only count up, such that the count may not be reversed and thereby, allow expired content to be displayed.

In yet another embodiment, a certificate containing the public key of the source device is sent to the display device prior to the above described process. A public key known to the display device may be used to verify the certificate. Preferably, the public key used to verify the certificate is built into the display device by the manufacturer of the display device. In this embodiment, the relative time dependent ticket (the checkpoint concatenated with the ticket) may be encrypted utilizing a private key of the source device. The encrypted relative time dependent ticket is then transmitted from the source device to the display device along with the watermarked content and the ticket. Thereafter, prior to the display device verifying the checkpoint, the display device decrypts the relative time dependent ticket utilizing a public key of the source device. In still yet another embodiment, the relative time dependent ticket may be signed (as is know in the art, by hashing the relative time dependent ticket and encrypting that hashed result) utilizing a private key of the source device. The resulting signature is sent along with the watermarked content, the relative time dependent ticket, and ticket to the display device. Thereafter, prior to the display device verifying the checkpoint, the display device verifies the signature on the relative time dependent ticket utilizing a public key of the source device.

Brief Description of the Drawings

The following are descriptions of embodiments of the present invention that when taken in conjunction with the following drawings will demonstrate the above noted features and advantages, as well as further ones. It should be expressly understood that the drawings are included for illustrative purposes and do not represent the scope of a present invention. The invention is best understood in conjunction with the accompanying drawings in which:

Fig. 1 shows a conventional system for the transmission of content;

Fig. 2 shows an illustrative communication network in accordance with an embodiment of the present invention; and

Fig. 3 shows details of an illustrative communication network in accordance with embodiment of the present invention wherein a source device provides content to a sink device.

Detailed Description of the Invention

Fig. 2 depicts an illustrative communication network 250 in accordance with an embodiment of the present invention. A source device 230, such as Set Top Box (STB), a Digital Video Disc (DVD), a Digital Video Cassette Recorder (DVCR), or another source of content, utilizes a transmission channel 260 to transmit content to a sink device 240. The transmission channel 260 may be a telephone network, a cable television network, a computer data network, a terrestrial broadcast system, a direct broadcast satellite network, some combination thereof, or some other suitable transmission system that is know in the art. As such, the transmission channel 260 may include RF transmitters, satellite transponders, optical fibers, coaxial cables, unshielded twisted pairs of wire, switches, in-line amplifiers, etc. The transmission channel 260 may also operate as a bi-directional transmission channel wherein signals may be transmitted from/to the source device 230, respectively, to/from the sink device 240. An additional transmission channel 261 may also be utilized between the source device 230 and the sink device 240. Typically, the transmission channel 260 is a wide-bandwidth channel that in addition to transmitting copy protection content (e.g., copy protection related messages), transmits copy protected content. The transmission channel 261 typically is a low-bandwidth channel that is utilized to transmit copy protection content.

The sink device 240 contains a memory 276 that is utilized for storing a checkpoint. The sink device 240 also contains a counter, such as a counter 272, that is utilized for generating the checkpoint. Preferably, the counter 272 should increment on a microsecond or better resolution as suitable for the application. The counter 272 should be free running. For instance, the counter 272 should count at all times that the sink device 240 is on. The bits of the counter 272 should employ non-volatile memory such as an electrically erasable programmable read-only memory (EEPROM) for the storage of the count. The counter 272 preferably is constructed to only count in one direction (e.g., up) and not in another direction (e.g., down). In a preferred embodiment, the counter 272 is driven by an inaccurate time source (e.g., inaccurate in terms of keeping time over hours, not necessarily over seconds), such as clock 274. The clock 274 is preferably unreliable so that drift with respect to time and temperature is also non-negligible. Over time, this has the effect of randomizing the count of a counter for each sink device of a population of sink devices. In addition, the counter 272 may be driven fast for a random period of time to initialize the counter 272 to a random number at the time of manufacture. All of the above, has an effect of further randomizing the counter 272. The counter 272 is also configured such that it is inaccessible to a user. Accordingly, the user may not reset the counter 272.

The checkpoint, in accordance with the present invention, is transmitted to the source device 230 utilizing at least one of the transmission channels 260, 261. The source device 230 utilizes the checkpoint to change the ticket such that the watermarked content may only be utilized (e.g., played) by a corresponding sink device as described in more detail below. In the event that the corresponding sink device, such as the sink device 240, receives the watermarked content, then the content may be provided to a device, such as a display device 265, for display thereon. Preferably, the display device 265 is integral to the sink device 240 such that the display device 265 is the final arbiter in determining whether the copy protected content may be utilized. It should be obvious that although the device is illustratively shown as the display device 265, in fact the device may be any known device that may be suitably utilized for the copy protected content. For instance, in a case wherein the copy protected content is audio content, the device may be the device that outputs the audio signal.

In one embodiment of the present invention, the content may be provided from the source device 230 in the form of a Moving Picture Experts Group (MPEG) compliant transport stream, such as an MPEG-2 compliant transport stream. However, the present invention is not limited to the protection of an MPEP-2 compliant transport stream. As a person skilled in the art would readily appreciate, the present invention may be suitably employed with any other data stream that is known in the art for transmitting content.

In another embodiment, the source device 230 may be a conditional access (CA) device. In this embodiment, the transmission channel 260 is a conditional access module bus.

Fig. 3 depicts details of an illustrative communication network 350 in accordance with an embodiment of the present invention. In the communication network 350, a source device 330 provides content including copy protected content to a sink device 340 over a transmission channel 360. As discussed above with regard to the transmission channel 260, the transmission channel 360 may be a wide bandwidth transmission channel that may also have a bi-directional capability, such as a CA module bus.

The sink device 340 contains a copy protection status determination circuit 370 for creating/storing a checkpoint (C) and for determining the copy protection status of received content. The copy protection status determination circuit 370 contains a counter 372 and a clock 374 for creating the checkpoint (C). The counter 372 preferably contains a large number of bits (e.g., 64 bits for a clock 374 that increments on a millisecond basis). Preferably, the counter 372 should have a total count cycle time (the time required for the counter 372 to reach a top count from a bottom count) longer than a useful life of the sink device 340 (e.g., ten years). The clock 374 is preferably randomized (e.g., unreliable such that drift with respect to time and temperature is non-negligible) as discussed above with regard to the clock 274 shown in Fig. 2. The counter 372 is configured such that it is inaccessible and has no reset function even in the event of a removal of power. As such, the counter 372 may contain non-volatile storage, such as programmable read-only memory (PROM), electrically erasable PROM (EEPROM), static random access memory (static-RAM), etc. Further, the copy protection status determination circuit 370 contains a memory device 376 for storing the checkpoint (C).

In operation, the source device 330 may request the checkpoint (C) from the sink device 340 prior to transmitting copy protected content. In alternate embodiments, the sink device 340 may transmit the checkpoint (C) to the source device 330 as a portion of a request for the source device 330 to begin transmission of copy protected content to the sink device 340. The sink device 340 may utilize either of the transmission channels 360, 361 for transmission of the request for copy protected content and/or for transmission of the checkpoint (C). However, in some embodiments of the present invention, the transmission channel 360 may be unidirectional and may only be utilized for the transmission of content to the sink device 340 from the source device 330. In these embodiments, the transmission channel 361 is utilized for the transmission of the checkpoint (C) from the sink device 340 to the source device 330. The transmission channel 361 may also be utilized for transmitting a request for copy protected content from the sink device 340 to the source device 330.

In an alternate embodiment, the transmission channel 360 has bi-directional capability and may be utilized for transmissions both to and from the source device 330, and to and from the sink device 340. In this embodiment, the transmission channel 361 may not be present or it may be utilized solely for the transmission of content requiring low bandwidth. For instance, the source device 330 may utilize the transmission channel 361 to transmit to the sink device 340 a request for the transmission of the checkpoint (C).

In one particular embodiment, the source device 330 is a conditional access (CA) device 330, the transmission channel 360 is a CA module bus 360, and the sink device 340 is a display device 340. Prior to the transmission of copy protected content, the CA device 330 transmits a request for a checkpoint (C) (e.g., the current count from the free running counter 372) from the display device 340. In response to the request, the display device 340 transmits the checkpoint (C) to the CA device 330 over the CA module bus 360. In addition to sending the checkpoint (C) to the CA device 330, the display device 340 saves the checkpoint (C) in the memory 376.

The CA device 330 contains a processor 314. The processor 314 utilizes a ticket and the checkpoint (C), received from the display device 340, to create a relative time dependent ticket (TDT) as discussed in more detail below. In one embodiment, the processor 314 may simply be a fixed hardware device that is configured for performing functions, such as mathematical functions, including a concatenation function, a one-way function, such as a hashing function, etc. In alternate embodiments, the processor 314 may be a microprocessor or a reconfigurable hardware device. What is intended by the term "relative time dependent ticket (TDT)" is that due to the randomization of the counter 372 as discussed above, the checkpoint (C) is not directly related to an absolute time amongst all sink devices. The checkpoint (C) is only related to a relative time of a given sink device such as the display device 340.

In one embodiment, the copy protected content is received via an input 305 as an audio/video (A/V) signal. Preferably, in this embodiment, the A/V signal contains a watermark (W) and a ticket (T). The watermark (W) and the ticket (T) are related as discussed with regard to the prior art (e.g., W = H(H(T))). Preferably, the watermark (W) is embedded into the copy protected content. In this way, removal of the watermark (W) from the copy protected content will result in the copy protected content becoming largely degraded. The ticket accompanies the content and is not embedded in it.

In an alternate embodiment, the copy protected content is read from a physical medium, such as a digital video disc (DVD). In this embodiment, the DVD may contain a physical mark (P) as described above. Further, content contained on the DVD (e.g., A/V content) has a watermark (W) embedded therein (e.g., watermarked content) such that removal of the watermark (W) from the A/V content results in the A/V content becoming largely degraded. In this embodiment, the physical mark (P), the ticket (T), and the watermark (W) are related as follows:

$$T = H(H(P)) \qquad (1)$$

$$W = H(H(T)) \qquad (2)$$

In any event, at the CA device 330, the checkpoint (C) is combined with the ticket (T), utilizing for instance concatenation and hashing functions. Thereby, a time dependent ticket (TDT) is created as follows:

$$TDT = H(T.C). \qquad (3)$$

The watermarked content, containing a watermark (W) embedded therein, the time dependent ticket (TDT), and the ticket (T), are then transmitted via the CA module bus 360 to the display device 340.

At the receiver 340, the copy protection status determination circuit 370 extracts the watermark (W) from the watermarked content. The copy protection status determination circuit 370 compares the watermark (W) and the ticket (T) in the usual way, as is known in the art (e.g., $W = H(H(T))$?).

In the event that the comparison does not pass (e.g., $W \neq H(H(T))$), then the content is discarded and any selected operation at the display device 340 (e.g., play, record, etc.) regarding the content is disabled. However, if the comparison does pass (e.g., $W = H(H(T))$), then the copy protection determination circuit 370 retrieves the stored checkpoint (C) from the memory 376 and combines the ticket (T) with the stored checkpoint (C), utilizing the same operation that was utilized at the source device 330 for creating the time dependent ticket (TDT). To this end, the receiver 340 comprises a processor 314' that is comparable to the processor 314 in the source device 330. For instance, concatenation and hashing functions may be utilized at the display device 340 for combining the ticket (T) with the stored checkpoint (C). A result of the combination is then compared to the time dependent ticket (TDT):

$$TDT = H(T.C)? \qquad (4)$$

In the event that the result does not equal the time dependent ticket (TDT), then the content is discarded and any selected operation at the display device (e.g., play, record, etc.) regarding the content is disabled. This may happen, for instance, in a case wherein an improper display device (e.g., a display device other than the display device that requested the content) has received the content. If the result does equal the time dependent ticket (TDT), then access to the content is enabled in accordance with the access granted by the ticket.

In a preferred embodiment, a further step is performed prior to the display device 340 having access to the copy protected content. Specifically, the checkpoint (C) stored in the memory 376 is compared to a current count of the (running) counter 372. In the event that the stored checkpoint (C) is within an allowable window of the current count from the counter 372 (e.g., within 24 hours of the count for some applications), then the display device 340 is provided with access to the copy protected content. What is an allowable window between the stored checkpoint (C) and the current count will depend on the nature of the desired content protection. For example, in one embodiment or for one particular type of content, the allowed window (the difference between the stored checkpoint (C) and the current count) may be short to ensure that the content is being transmitted and received in real time. In another embodiment or for another type of content, the allowed window may be longer (e.g., months or years) to allow for storage of the content for later playback.

If the checkpoint (C) has expired (e.g., not within the allowed window), then the checkpoint (C) is erased and the display device 340 is not provided with access to the copy protected content. As is readily ascertained by a person of ordinary skill in the art, the comparison of the checkpoint (C) to the current count may be performed any time prior to the display device having access to the copy protected content. In a preferred embodiment, the checkpoint (C) is compared to the current count prior to the comparison of the watermark (W) to the ticket (T).

It should be clear that a trusted source should be utilized to create the recorded content or the real time transmitted content (e.g., received over the input 305). A CA device, such as the CA device 330, which is inherently designed to be tamper resistant is an example of a trusted real time source. In this case, it may be assumed that the CA device 330 decrypts the watermarked content so that prior to the watermarked contents arrival at the CA device 330, the watermarked content cannot be recorded.

In a case wherein the ticket (T) does not properly compare to the watermark (W), or some other portion of the copy protection status determination process fails, the copy protected content is discarded. In addition, when the copy protection status determination process fails, no operation regarding the copy protected content is enabled at the display device 340.

In accordance with the present invention, a checkpoint (C) from a counter of a given display device is in effect unique. Accordingly, the copy protected content transmitted by the CA device 330 may not be distributed to a display device other than the display device that sent the checkpoint (C). In addition, by comparing the checkpoint (C) to the count of the counter 372, the copy protected content may be restricted to being played within a time, as determined by the window of time as discussed above.

In yet another embodiment, a private/public key system, as is known by a person of ordinary skill in the art, is utilized to further secure the copy protected content in accordance with the present invention. In accordance with this embodiment, the display device 340 has a public key that is trusted e.g., secure for example by being installed in part of the display device hardware, such as stored in the memory 376. The public key corresponds to a private key of the manufacturer of the display device 340 and is stored, for instance, in a memory 322 at the CA device 330. The private key is utilized to sign certificates of each CA device manufacturer, as is known in the art.

In operation, when the CA device 330 is connected to the display device 340 via the CA module bus 360, a certificate containing the CA device 330 public key is sent to the display device 340. Once the certificate containing the public key of the CA device 330 is verified by the display device 340, as is known in the art, the public key of this CA device 330 is stored at the display device 340. Thereafter, the CA device 330 may digitally sign the time dependent ticket (TDT). For instance, the time dependent ticket (TDT) may be hashed and the result may be encrypted by the private key of the CA device 330 to form a signature. The signature is sent from the CA device 330 to the display device 340 together with the watermarked content, the ticket, and the time dependent ticket (TDT). At the display device 340, the signature is verified utilizing the public key of the CA device 330 and thereafter, the time dependent ticket (TDT) and checkpoint (C) are utilized as described above.

In yet another embodiment, the time dependent ticket (TDT) may be encrypted utilizing the private key of the CA device 330. The encrypted time dependent ticket (TDT) is then transmitted from the CA device 330 to the display device 340 along with the watermarked content and the ticket (T). Thereafter, prior to the display device 340 verifying the checkpoint (C), the display device 340 decrypts the time dependent ticket (TDT) utilizing the public key of the CA device 330. Thereafter, the time dependent ticket (TDT) may be utilized as discussed above.

An illustrative protocol for use of a checkpoint and a private/public key system in accordance with an embodiment of the present invention is described below. In accordance with the present invention, after a CA device is connected to a display device, the CA device sends a certificate containing the CA device public key to the display device. The display device verifies the certificate utilizing the embedded public key of the manufacturer and stores the verified public key of the CA device. In response to a request for copy protected content from the display device, the CA device requests a checkpoint (C) from the display device. The display device sends the checkpoint (C) to the CA device and also stores a copy of the checkpoint (C) locally (e.g., at the display device). The CA device combines the checkpoint (C) with the ticket (T) utilizing concatenation and hashing functions to produce a time dependent ticket (TDT). The CA device encrypts the time dependent ticket (TDT) utilizing the CA device private key. The encrypted time dependent ticket (TDT) is then sent to the display device along with the watermarked content and the ticket (T). The display device compares the stored checkpoint (C) with the current state of a counter to determine if the checkpoint (C) is within an allowable window of time of the current state of the counter. If the stored checkpoint (C) is not within the allowable window of time of the current state of the counter, then access to the content is disabled. If the stored checkpoint (C) is within the allowable window, then the display device utilizes the public key of the CA device to decrypt the time dependent ticket (TDT). The display device combines the ticket (T) with the stored checkpoint (C) utilizing concatenation and hashing functions and compares a result to the time dependent ticket (TDT). If the result is not equal to the time dependent ticket (TDT), then access to the content is disabled. If the result is equal to the time dependent ticket (TDT), the ticket and watermark are compared in the usual way. If step 480 fails (e.g., $W \neq H(H(T))$), then in step 485, access to the content is disabled. If the ticket and the watermark do no correspond, (e.g., $W = H(H(T))$), access to the content is enabled (e.g., the content may be displayed).

The following embodiments of the invention overcome the disadvantages of the prior art. A display device is provided that is the final arbiter in deciding whether to display the protected content. In this way, the display device is the gatekeeper that disallows recordings that are made and played back on non-compliant players/recorders. A further embodiment provides a method of transmitting copy protected copy-never content that will prevent a pirate from making copies that will display on a compliant display device. A ticket is created that is unique to a particular display device so that copy protected content will only play on the particular display device. A still further embodiment creates a ticket that is inspected by the display device to decide whether the content is being transmitted in real time. A time dependent ticket is created that is checked by a display device to determine if content has expired or aged beyond an allowable window of time from a checkpoint. Another embodiment of the invention uses a relative time reference configured such that each display device has a different relative time reference.

It should be noted that the above-mentioned embodiments illustrate rather than limit the invention, and that those skilled in the art will be able to design many alternative embodiments without departing from the scope of the appended claims. In the claims, any reference signs placed between parentheses shall not be construed as limiting the claim. The word "comprising" does not exclude the presence of other elements or steps than those listed in a claim. Another embodiment of the invention can be implemented by means of hardware comprising several distinct elements, and by means of a suitably programmed computer. In a device claim enumerating several means, several of these means can be embodied by one and the same item of hardware.

CLAIMS:

1. A method of protecting content transmitted as a stream of data, the method comprising the steps of:

determining a checkpoint at a receiving device (240);

calculating, at a source device (230), a time dependent ticket utilizing the checkpoint, wherein a watermark, a ticket, and the checkpoint together indicate a copy protection status of the content;

transmitting said stream of data, said watermark, said ticket, and said time dependent ticket to said receiving device (240); and

comparing said time dependent ticket to a stored checkpoint at said receiving device (240).

2. The method of claim 1, wherein said step of calculating said time dependent identifier comprises the steps of:

combining said checkpoint with said ticket, and

calculating a one-way operation on said combined checkpoint and ticket.

3. The method of claim 2, further comprising the step of selecting said one-way function to be a hashing function.

4. The method of claim 1, further comprising the step of comparing, at said receiving device (240), said ticket and said watermark to determine the copy protection status of the content if said time dependent ticket compares to said stored checkpoint.

5. The method of claim 1, wherein said checkpoint is a checkpoint from a receiver counter (272).

6. The method of claim 5, wherein said receiver counter (272) is randomized.

7. The method of claim 5, wherein the step of comparing said time dependent ticket further comprises the step of comparing said stored checkpoint to a current count from said receiver counter (272).

8. The method of claim 1, wherein said step of calculating said time dependent ticket further comprises the step of signing said time dependent ticket with a private key of said source device (230), and wherein said step of comparing said time dependent ticket further comprises the step of verifying the signature using a public key of said source device (230).

9. A copy protection system for protecting content wherein a ticket and a watermark indicates a copy protection status of said content, the system comprising:

a source device (330) configured to calculate a time dependent ticket using a checkpoint and a one-way function, and to provide a data stream containing said content, said ticket, a watermark, and said time dependent ticket; and

a display device (340) configured to produce said checkpoint, configured to receive said data stream, and configured to compare said time dependent ticket to said checkpoint using said ticket and said one-way function.

10. The system of claim 9, wherein said display device (340) is further configured to compare said ticket to said watermark and to display said content if said time dependent ticket compares to said checkpoint.

11. The system of claim 9, wherein said display device (340) comprises a counter (372) and wherein said checkpoint is a checkpoint from said counter (372).

12. The system of claim 11, wherein said display device (340) is further configured to randomize said counter (372).

13. The system of claim 11, wherein said display device (340) is further configured to compare said checkpoint to a current count from said counter (372) prior to displaying said content.

14. A source device (330) for protecting content wherein a ticket and a watermark indicate a copy protection status of the content, said source device (330) comprising:

a reader device configured to read watermarked content from a physical medium and configured to read a physical mark from said physical medium; and

a processor (314) configured to receive a checkpoint, configured to calculate said ticket using said physical mark and a one-way function, configured to calculate a time dependent ticket using said ticket, said checkpoint, and said one-way function, and configured to provide to a receiver (340) a data stream containing said watermarked content, said ticket, and said time dependent ticket.

15. A display device (340) for receiving data containing watermarked content and a ticket, wherein said ticket and watermark together indicate a copy protection status of the content, said display device comprising:

a counter (372) configured to provide a checkpoint and a current time reference; and

a processor (314'), wherein if said checkpoint is contained within a time window determined by said current time reference, said (314') processor is configured to:

receive a time dependent ticket and said data,

combine said ticket with said checkpoint to produce a first result,

perform a one-way function on said first result to produce a second result, and compare said second result to said time dependent ticket, wherein said display device (340) is further configured to display said data if said second results compares to said time dependent ticket.

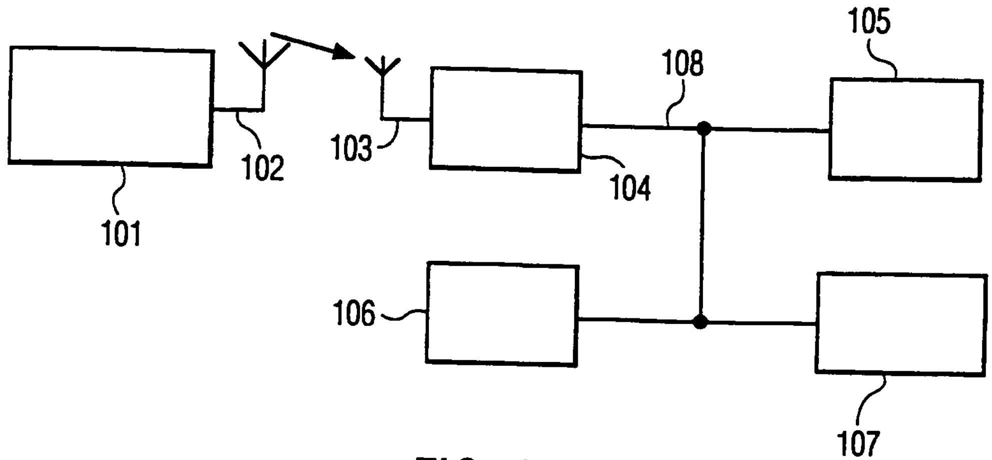

105
108
102
103
104
101
106
107

FIG. 1

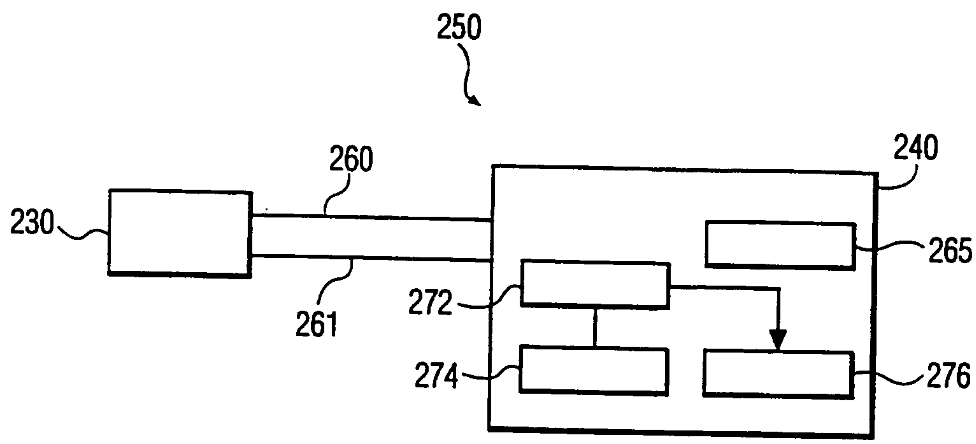

250
240
260
230
265
272
261
274
276

FIG. 2

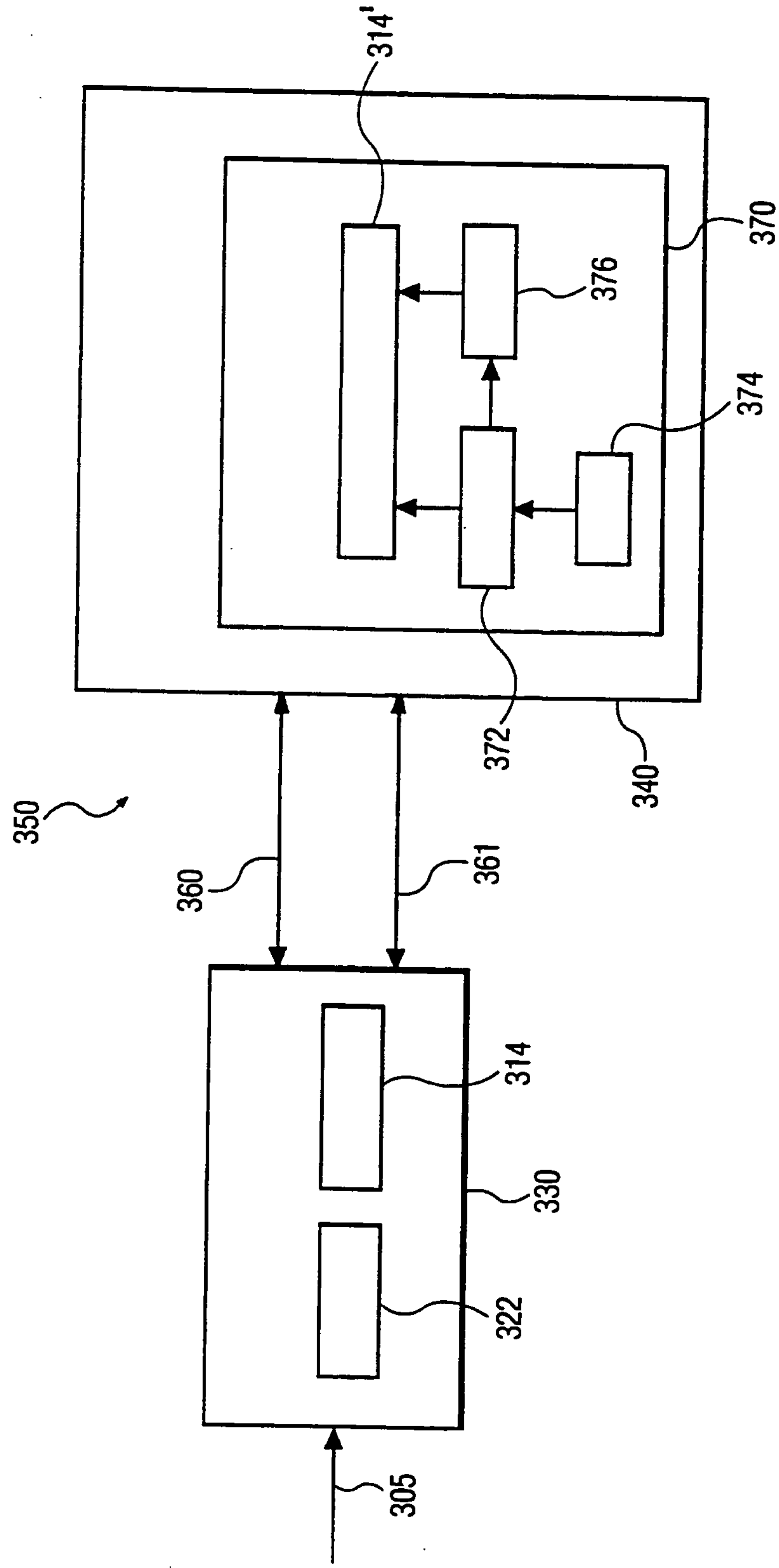


FIG. 3

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04N 7/30

A2

(11) International Publication Number: WO 00/05898

(43) International Publication Date: 3 February 2000 (03.02.00)

(21) International Application Number: PCT/US99/16638

(22) International Filing Date: 21 July 1999 (21.07.99)

(30) Priority Data:

60/093,860	23 July 1998 (23.07.98)	US
09/169,829	11 October 1998 (11.10.98)	US

(63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Applications

US	60/093,860 (CIP)
Filed on	23 July 1998 (23.07.98)
US	09/169,829 (CIP)
Filed on	11 October 1998 (11.10.98)

(71) Applicant (*for all designated States except US*): OPTIVISION, INC. [US/US]; 3450 Hillview Avenue, Palo Alto, CA 94304 (US).

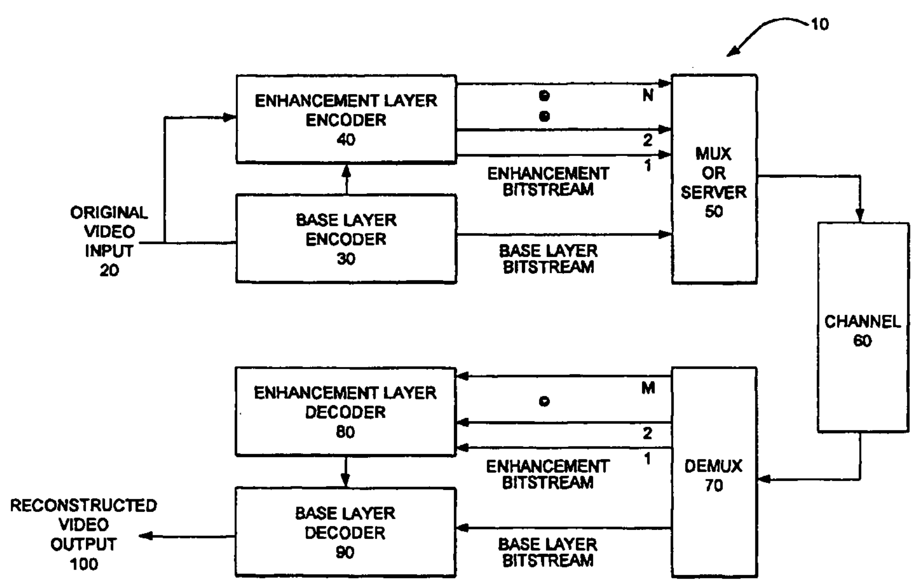
(72) Inventor; and
(75) Inventor/Applicant (*for US only*): LI, Weiping [US/US]; 159 California Avenue, J103, Palo Alto, CA 94306 (US).

(74) Agent: DAVIS, Paul; Wilson Sonsini Goodrich & Rosati, 650 Page Mill Road, Palo Alto, CA 94304-1050 (US).

(81) 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, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, 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
Without international search report and to be republished upon receipt of that report.

(54) Title: SCALABLE VIDEO CODING AND DECODING



(57) Abstract

A video encoding method and apparatus for adapting a video input to a bandwidth of a transmission channel of a network that includes determining the number N enhancement layer bitstreams capable of being adapted to the bandwidth of the transmission channel of a network. A base layer bitstream is encoded from the video input wherein a plurality of enhancement layer bitstreams are encoded from the video input. The enhancement layer bitstreams are based on the base layer bitstream, wherein the plurality of enhancement layer bitstreams complements the base layer bitstream and the base layer bitstream and N enhancement layer bitstreams are transmitted to the network.

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	**ML**	Mali	**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**MN**	Mongolia	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MR**	Mauritania	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MW**	Malawi	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MX**	Mexico	**US**	United States of America
CA	Canada	**IT**	Italy	**NE**	Niger	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NL**	Netherlands	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NO**	Norway	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NZ**	New Zealand	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**PL**	Poland		
CM	Cameroon	**KR**	Republic of Korea	**PT**	Portugal		
CN	China	**KZ**	Kazakstan	**RO**	Romania		
CU	Cuba	**LC**	Saint Lucia	**RU**	Russian Federation		
CZ	Czech Republic	**LI**	Liechtenstein	**SD**	Sudan		
DE	Germany	**LK**	Sri Lanka	**SE**	Sweden		
DK	Denmark	**LR**	Liberia	**SG**	Singapore		
EE	Estonia						

SCALABLE VIDEO CODING AND DECODING

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a method and apparatus for the scaling of data signals the bandwidth of the transmission channel; and more particularly to a scalable video method and apparatus for coding video such that the received video is adapted to the bandwidth of the transmission channel.

Description of Related Art

Signal compression in the video arena has long been employed to increase the bandwidth of either the generating, transmitting, or receiving device. MPEG - an acronym for Moving Picture Experts Group - refers to the family of digital video compression standards and file formats developed by the group. For instance, the MPEG-1 video sequence is an ordered stream of bits, with special bit patterns marking the beginning and ending of a logical section.

MPEG achieves high compression rate by storing only the changes from one frame to another, instead of each entire frame. The video information is then encoded using a technique called DCT (Discrete Cosine Transform) which is a technique for representing a waveform data as a weighted sum of cosines. MPEG use a type of lossy compression wherein some data is removed. But the diminishment of data is generally imperceptible to the human eye. It should be noted that the DCT itself does not lose data; rather, data compression technologies that rely on DCT approximate some of the coefficients to reduce the amount of data.

The basic idea behind MPEG video compression is to remove spatial redundancy within a video frame and temporal redundancy between video frames. The DCT-based (Discrete Cosine Transform) compression is used to reduce spatial redundancy and motion compensation is used to exploit temporal redundancy. The images in a video stream usually do not change much within small time intervals.

Thus, the idea of motion-compensation is to encode a video frame based on other video frames temporally close to it.

A video stream is a sequence of video frames, each frame being a still image. A video player displays one frame after another, usually at a rate close to 30 frames per second. Macroblocks are formed, each macroblock consists of four 8 x 8 luminance blocks and two 8 x 8 chrominance blocks. Macroblocks are the units for motion-compensated compression, wherein blocks are basic unit used for DCT compression. Frames can be encoded in three types: intra-frames (I-frames), forward predicted frames (P-frames), and bi-directional predicted frames (B-frames).

An I-frame is encoded as a single image, with no reference to any past or future frames. Each 8 x 8 block is encoded independently, except that the coefficient in the upper left corner of the block, called the DC coefficient, is encoded relative to the DC coefficient of the previous block. The block is first transformed from the spatial domain into a frequency domain using the DCT (Discrete Cosine Transform), which separates the signal into independent frequency bands. Most frequency information is in the upper left corner of the resulting 8 x 8 block. After the DCT coefficients are produced the data is quantized, i.e. divided or separated. Quantization can be thought of as ignoring lower-order bits and is the only lossy part of the whole compression process other than sub-sampling.

The resulting data is then run-length encoded in a zig-zag ordering to optimize compression. The zig-zag ordering produces longer runs of 0's by taking advantage of the fact that there should be little high-frequency information (more 0's as one zig-zags from the upper left corner towards the lower right corner of the 8 x 8 block).

A P-frame is encoded relative to the past reference frame. A reference frame is a P- or I-frame. The past reference frame is the closest preceding reference frame. A P-macroblock is encoded as a 16 x 16 area of the past reference frame, plus an error term.

To specify the 16 x 16 area of the reference frame, a motion vector is included. A motion vector (0, 0) means that the 16 x 16 area is in the same position as the macroblock we are encoding. Other motion vectors are generated are relative to that position. Motion vectors may include half-pixel values, in which case pixels are averaged. The error term is encoded using the DCT, quantization, and run-length encoding. A macroblock may also be skipped which is equivalent to a (0, 0) vector and an all-zero error term.

A B-frame is encoded relative to the past reference frame, the future reference frame, or both frames.

A pictorial view of the above processes and techniques in application are depicted in prior art Fig. 15, which illustrates the decoding process for a SNR scalability. Scalable video coding means coding video in such a way that the quality of a received video is adapted to the bandwidth of the transmission channel. Such a coding technique is very desirable for transmitting video over a network with a time-varying bandwidth.

SNR scalability defines a mechanism to refine the DCT coefficients encoded in another (lower) layer of a scalable hierarchy. As illustrated in prior art Fig. 15, data from two bitstreams is combined after the inverse quantization processes by adding the DCT coefficients, Until the dat is combined, the decoding processes of the two layers are independent of each other.

The lower layer (base layer) is derived from the first bitstream and can itself be either non-scalable, or require the spatial or temporal scalability decoding process, and hence the decoding of additional bitstream, to be applied. The enhancement layer, derived from the second bitstream, contains mainly coded DCT coefficients and a small overhead.

In the current MPEG-2 video coding standard, there is an SNR scalability extension that allows two levels of scalability. MPEG achieves high compression rate by storing only the changes from one frame to another, instead of each entire frame. There are at least two disadvantages of employing the MPEG-2 standard for encoding video data. One disadvantage is that the scalability granularity is not fine enough, because the MPEG-2 process is an all or none method. Either the receiving device can receive all of the data from the base layer and the enhancement layer or only the data from the base layer bitstream. Therefore, the granularity is not scalable. In a network environment, more than two levels of scalability are usually needed.

Another disadvantage is that the enhancement layer coding in MPEG-2 is not efficient. Too many bits are needed in the enhancement layer in order to have a noticeable increase in video quality.

The present invention overcomes these disadvantages and others by providing, among other advantages, an efficient scalable video coding method with increased granularity.

SUMMARY OF THE INVENTION

The present invention can be characterized as a scalable video coding means and a system for encoding video data, such that quality of the final image is gradually improved as more bits are received. The improved quality and scalability are achieved by a method wherein an enhancement layer is subdivided into layers or levels of bitstream layers. Each bitstream layer is capable of carrying information complementary to the base layer information, in that as each of the enhancement layer bitstreams are added to the corresponding base layer bitstreams the quality of the resulting images are improved.

The number N of enhancement layers is determined or limited by the network that provides the transmission channel to the destination point. While the base layer bitstream is always transmitted to the destination point, the same is not necessarily true for the enhancement layers. Each layer is given a priority coding and transmission is effectuated according to the priority coding. In the event that all of the enhancement layers cannot be transmitted the lower priority coded layers will be omitted. The omission of one or more enhancement layers may be due to a multitude of reasons.

For instance, the server which provides the transmission channel to the destination point may be experiencing large demand on its resources from other users, in order to try and accommodate all of its users the server will prioritize the data and only transmit the higher priority coded packets of information. The transmission channel may be the limiting factor because of the bandwidth of the channel, i.e. Internet access port, Ethernet protocol, LAN, WAN, twisted pair cable, co-axial cable, etc. or the destination device itself, i.e. modem, absence of an enhanced video card, etc. may not be able to receive the additional bandwidth made available to it. In these instances only M number (M is an integer number = 0, 1, 2, . . .) of enhancement layers may be received, wherein N number (N is an integer number = 0, 1, 2, . . .) of enhancement layers were generated at the encoding stage, $M \leq N$.

To achieve these and other advantages and in accordance with the purpose of the present invention, as embodied and broadly described, the scalable video method and apparatus according to one aspect of the invention includes a video encoding method for adapting a video input to a bandwidth of a transmission channel of a network, the method includes determining the number N of enhancement layer bitstreams capable of being adapted to the bandwidth of the transmission channel of the network. Encoding a base layer bitstream from the video input is then performed and encoding N number of enhancement layer bitstreams from the video input based on the base layer bitstream, wherein the plurality of enhancement layer bitstreams complements the base layer bitstream. The base layer bitstream and the N enhancement layer bitstreams are then provided to the network.

According to another aspect of the present invention, a video decoding method for adapting a video input to a bandwidth of a transmission channel of a network includes, determining number M of enhancement layer bitstreams of said video input capable of being received from said transmission channel of said network. Decoding a base layer bitstream from received video input and decoding M number of enhancement layer bitstreams from the received video input based on the base layer bitstream, wherein the M received enhancement layer bitstreams complements the base layer bitstream. Then reconstructing the base layer bitstream and N enhancement layer bitstreams.

According to still another aspect of the present invention, a video decoding method for adapting a video input to a bandwidth of a receiving apparatus, the method includes demultiplexing a base layer bitstream and at least one of a plurality of enhancement layer bitstreams received from a network, decoding the base layer bitstream, decoding at least one of the plurality of enhancement layer bitstreams based on generated base layer bitstream, wherein the at least one of the plurality of enhancement layer bitstreams enhances the base layer bitstream. Then reconstructing a video output.

According to a further aspect of the present invention, a video encoding method for encoding enhancement layers based on a base layer bitstream encoded from a video input, the video encoding method includes, taking a difference between an original DCT coefficient and a reference point and dividing the difference between the original DCT coefficient and the reference point into N bit-planes.

According to a still further aspect of the present invention, a method of coding motion vectors of a plurality of macroblocks, includes determining an average motion vector from N motion vectors for N macroblocks, utilizing the determined average motion vector as the motion vector for the N macroblocks, and encoding 1/N motion vectors in a base layer bitstream.

Additional features and advantages of the invention will be set forth in the description which follows, and in part will be apparent from the description, or may be learned by practice of the invention. The aspects and other advantages of the invention will be realized and attained by the structure particularly pointed out in the written description and claims hereof as well as the appended drawings.

It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory and are intended to provide further explanation of the invention as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are included to provide a further understanding of the invention and are incorporated in and constitute a part of this specification, illustrate embodiments of the invention and together with the description serve to explain the principles of the invention. In the drawings:

Fig. 1 illustrates a flow diagram of the scalable video encoding method of the present invention;

Fig. 2A illustrates conventional probability distribution of DCT coefficient values;

Fig. 2B illustrates conventional probability distribution of DCT coefficient residues;

Fig. 3A illustrates the probability distribution of DCT coefficient values of the present invention;

Fig. 3B illustrates the probability distribution of DCT coefficient residues of the present invention;

Figs. 3C and 3D illustrates a method for taking a difference of a DCT coefficient of the present invention;

Fig. 5 illustrates a flow diagram for finding the maximum number of bit-planes in the DCT differences of a frame of the present invention;

Fig. 6 illustrates a flow diagram for generating (RUN, EOP) Symbols of the present invention;

Fig. 7 Illustrates a flow diagram for encoding enhancement layers of the present invention;

Fig. 8 illustrates a flow diagram for encoding (RUN, EOP) symbols and sign_enh values of one DCT block of one bit-plane;

Fig. 9 illustrates a flow diagram for encoding a sign_enh value of the present invention;

Fig. 10 illustrates a flow diagram for adding enhancement difference to a DCT coefficient of the present invention;

Fig. 11 illustrates a flow diagram for converting enhancement difference to a DCT coefficient of the present invention;

Fig. 12 illustrates a flow diagram for decoding enhancement layers of the present invention;

Fig. 13 illustrates a flow diagram for decoding (RUN, EOP) symbols and sign_enh values of one DCT block of one bit-plane;

Fig. 14 illustrates a flow diagram for decoding a sign_enh value; and

Fig. 15 illustrates a prior a conventional SNR scalability flow diagram.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings.

Fig. 1 illustrates the scalable video diagram 10 of an embodiment of the present invention. The original video input 20 is encoded by the base layer encoder 30 in accordance with the method of represent by flow diagram 400 of Fig. 4. A DCT coefficient OC and its corresponding base layer quantized DCT coefficient QC are generated and a difference determined pursuant to steps 420 and 430 of Fig. 4. The difference information from the base layer encoder 30 is passed to the enhancement layer encoder 40 that encodes the enhancement information.

The encoding of the enhancement layer encoder is performed pursuant to methods 500 - 900 as depicted in Figs. 5 - 10, respectively and will be briefly described. The bitstream from the base layer encoder 30 and the N bitstreams from the enhancement layer encoder 40 are capable of being sent to the transmission channel 60 by at least two methods.

In the first method all bitstreams are multiplexed together by multiplexor 50 with different priority identifiers, e.g., the base layer bitstream is guaranteed, enhancement bitstream layer 1 provided by enhancement layer encoder 40 is given a higher priority than enhancement bitstream layer 2. The prioritization is continued until all N (wherein N is an integer from 0, 1, 2, . . .) of the bitstreams layers are prioritized. Logic in the encoding layers 30 or 40 in negotiation with the network and intermediated devices determine the number N of bitstream layers to be generated.

The number of bitstream layers generated is a function of the total possible bandwidth of the transmission channel 60, i.e. Ethernet, LAN, or WAN connections (this list is not intended to exhaustive but only representation of potential limiting devices and/or equipment), and the network and other intermediate devices. The number of bitstream layers M (wherein M is an integer and $M \leq N$) reaching the destination point 100 can be further limited by not just the physical constraints of the intermediate devices but the congestion on the network, thereby necessitating the dropping of bitstream layers according to their priority.

In a second method the server 50 knows the transmission channel 60 condition, i.e. congestion and other physical constraints, and selectively sends the bitstreams to the channel according to the priority identifiers. In either case, the destination point 100 receives the bitstream for the base layer and M bitstreams for the enhancement layer, where $M \leq N$.

The bitstreams M are sent to the base layer 90 and enhancement layer 80 decoders after being demultiplexed by demultiplexor 70. The decoded enhancement information from the enhancement layer decoder is passed to the base layer decoder to composite the reconstructed video output 100. The decoding of the multiplexed bitstreams are accomplished pursuant to the methods and algorithms depicted in flow diagrams 1100 - 1400 of Figs. 11 - 14, respectively.

The base layer encoder and decoder are capable of performing logic pursuant to the MPEG-1, MPEG-2, or MPEG-4 (Version-1) standards that are hereby incorporated by reference into this disclosure.

Taking Residue with Probability Distribution Preserved

A detailed description of the probability distribution residue will now be made with reference to Figs 2A - 3B

In the current MPEG-2 signal-to-noise ratio (SNR) scalability extension, a residue or difference is taken between the original DCT coefficient and the quantized DCT coefficient. Fig. 2A illustrates the distribution of a residual signal as a DCT coefficient. In taking the residue small values have higher probabilities and large values have smaller probabilities. The intervals along the horizontal axis represent quantization bins. The dot in the center of each interval represents the quantized DCT coefficient. Taking the residue between the original and the quantized DCT coefficient is equivalent to moving the origin to the quantization point.

Therefore, the probability distribution of the residue becomes that as shown in Figure 2B. The residue from the positive side of Fig. 2A has a higher probability of being negative than positive and the residue taken from the negative side of the Fig. 2A has a higher probability of being positive than negative. The result is that the probability distribution of the residue becomes almost uniform. Thus making coding the residue more difficult.

A vastly superior method is to generate a difference between the original and the lower boundary points of the quantized interval as shown in Fig. 3A and Fig. 3B. In this method, the residue is taken from the positive side of Fig. 2A remains positive and the residue from the negative side of Fig. 2A remains negative. Taking the residue is equivalent to moving the origin to the reference point as illustrated in Fig. 3A. Thus, the probability of the residue becomes as shown in Fig. 3B. This method preserves the shape of the original non-uniform distribution. Although the dynamic range of the residue taken in such a manner seems to be twice of that depicted in Fig. 2B, their is no longer a need to code the sign, i.e. - or +, of the residue. The sign of the residue is encoded in the base layer bitstream corresponding the enhancement layer, therefore this redundancy is eliminated and bits representing the sign are thus saved. Therefore, there is only a need to code the magnitude that still has a nonuniform distribution.

Bit plane coding of residual DCT coefficients

After taking residues of all the DCT coefficients in an 8 x 8 block, bit plane coding is used to code the residue. In bit-plane coding method the bit-plane coding method considers each residual DCT coefficient as a binary number of several bits instead of as a decimal integer of a certain value as in the run-level coding method. The bit-plane coding method in the present invention only replaces runlevel coding part. Therefore, all the other syntax elements remain the same.

An example of and description of the bit-plane coding method will now be made, wherein 64 residual DCT coefficients for an Inter-block and 63 residual DCT coefficients for an Intra-block (excluding the Intra-DC component that is coded using a separate method) are utilized for the example. The 64 (or 63) residual DCT coefficients are ordered into a one-dimensional array and at least one of the residual coefficients is non-zero. The bit-plane coding method then performs the following steps.

The maximum value of all the residual DCT coefficients in a frame is determined and the minimum number of bits, N, needed to represent the maximum value in the binary format is also determined. N is the number of bitplanes layers for this frame and is coded in the frame header.

Within each 8 x 8 block is represent every one of the 64 (or 63) residual DCT coefficients with N bits in the binary format and there is formed N bit-planes or layers or levels. A bit-plane is defined as an array of 64 (or 63) bits, taken one from each residual DCT coefficient at the same significant position.

The most significant bit-plane is determined with at least one non-zero bit and then the number of all-zero bit-planes between the most significant bit-plane determined and the Nth one is coded. Then starting from the most significant bit plane (MSB plane), 2-D symbols are formed of two components: (a) number of consecutive O's before a I (RUN), (b) whether there are any l's left on this bit plane, i.e. End-Of-Plane (EOP). If a bit-plane after the MSB plane contains all O's, a special symbol

ALL-ZERO is formed to represent an all-zero bit-plane. Note that the MSB plane does not have the all-zero case because any all-zero bit-planes before the MSB plane have been coded in the previous steps.

Four 2-D VLC tables are used, wherein the table VT-C-Table-0 corresponds to the MSB plane; table VLC-Table- I corresponds to the second MSB plane; table VLC-Table-2 corresponds to the third MSB plane; and table VLC-Table-3 corresponds to the fourth MSB and all the lower bit planes. For the ESCAPE cases, RUN is coded with 6 bits, EOP is coded with 1 bit. Escape coding is a method to code very small probability events which are not in the coding tables individually.

An example of the above process will now follow. For illustration purposes, we will assume that the residual values after the zigzag ordering are given as follows and N = 6: The following representation is thereby produced.

10, 0, 6, 0, 0, 3, 0, 2, 2, 0, 0, 2, 0, 0, 1, 0, ... 0, 0

The maximum value in this block is found to be 10 and the minimum number of bits to represent 10 in the binary format (1010) is 4. Therefore, two all-zero bit-planes before the MSB plane are coded with a code for the value 2 and the remaining 4 bit-planes are coded using the (RUN, EOP) codes. Writing every value in the binary format using 4 bits, the 4 bit-planes are formed as follows:

1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0, 0 (MSB-plane)
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0, 0 (Second MSB-plane)
1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0,0, 0 (Third MSB-plane)
0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,0, 0 (Fourth MSB-plane or LSB-plane)

Converting the bits of each bit-plane into (RUN, EOP) symbols results in the following:

(0, 1)	(MSB-plane)
(2, 1)	(Second MSB-plane)
(0, 0), (1,O), (2,O), (1,O), (0, 0), (2, 1)	(Third MSB-plane)

(5, 0), (8, 1) (Fourth MSB-plane or LSB-plane)

Therefore, there are 10 symbols to be coded using the (RUN, EOP) VLC tables. Based on their locations in the bit-planes, different VLC tables are used for the coding. The enhancement bitstream using all four bitplanes looks as follows:

code leading-all-zero(2)

code msb(O, 1)

code msb-1(2,1)

code-msb-2(0,0), code_msb-2(1,0), code-msb-2(2,O), code-msb-2(1,0), code-msb-2(0,0), code-msb-2(2, 1) code_msb-3(5,0), code_msb-3(8, 1).

In an alternative embodiment, several enhancement bitstreams may be formed from the four bit-planes, in this example from the respective sets comprising one or more of the four bit-planes.

Motion Vector Sharing

In this alternative embodiment of the present invention motion vector sharing is capable of being utilized when the base layer bitstream exceeds a predetermined size or more levels of scalability are needed for the enhancement layer. By lowering the number of bits required for coding the motion vectors in the base layer the bandwidth requirements of the base layer bitstream is reduced. In base layer coding, a macroblock (16 x l6 pixels for the luminance component and W pixels for each chron-luminance components) of the current frame is compared with the previous frame within a search range. The closest match in the previous frame is used as a prediction of the current macroblock. The relative displacement of the prediction to the current macroblock, in the horizontal and vertical directions, is called a motion vector.

The difference between the current macroblock and it's prediction is coded using the DCT coding. In order for the decoder to reconstruct the current macroblock, the motion vector has to be coded in the bitstream. Since there is a fixed number of bits for coding a frame, the more bits spent on coding the motion vectors results in fewer bits for coding the motion compensated differences. Therefore, it is desirable to lower the number of bits for coding the motion vectors and leave more bits for coding the differences between the current macroblock and its prediction.

For each set of 2 x 2 motion vectors, the average motion vector can be determined and used for the four macroblocks. In order to not change the syntax of the base layer coding, four macroblocks are forced to have the identical motion vectors. Since only one out four motion vectors is coded in the bitstream, the amount of bits spent on motion vector coding is reduced, therefore, there are more bits available for coding the differences. The cost for pursuing such a method is that the four macroblocks, which share the same motion vector may, not get the best matched prediction individually and the motion compensated difference may have a larger dynamic range, thus necessitating more bits to code the motion vector.

For a given fixed bitrate, the savings from coding one out of four motion vectors may not compensate the increased number of bits required to code the difference with a larger dynamic range. However, for a time varying bitrate, a wider dynamic range for the enhancement layer provides more flexibility to achieve the best possible usage of the available bandwidth.

<u>**Coding Sign Bits**</u>

In an alternative embodiment of the present invention, if the base layer quantized DCT coefficient is non-zero, the corresponding enhancement layer difference will have the same sign as the base layer quantized DCT. Therefore, there is no need to code the sign bit in the enhancement layer.

Conversely, if the base layer quantized DCT coefficient is zero and corresponding enhancement layer difference is non-zero, a sign bit is placed into enhancement layer bitstream immediately after the MSB of the difference.
An example of the above method will now follow.

Difference of a DCT block after ordering

- 10, 0, 6, 0, 0, 3, 0, 2, 2, 0, 0, 2, 0, 0, 1, 0, ...0, 0

Sign indications of the DCT block after ordering

- 3, 3, 3, 3, 2, 0, 3, 3, 1, 2, 2, 0, 3, 3, 1, 2, ... 2, 3

- 0: base layer quantized DCT coefficient = 0 and difference >0

- 1: base layer quantized DCT coefficient = 0 and difference <0

- 2: base layer quantized DCT coefficient = 0 and difference =0

- 3: base layer quantized DCT coefficient = 0.

In this example, the sign bits associated with values 10, 6, 2 don't need to be coded and the sign bits associated with 3, 2, 2, 1 are coded in the following way:

Code(All Zero)
code (All Zero)
code(0,1)
code(2,1)
code(0,0),code(1,0),code(2,0),0,code(1,0),code(0,0),1,code(2,1),0
code(5,0),code(8,1),1

For every DCT difference, there is a sign indication associated with it. There are four possible cases. In the above coding 0, 1, 2, and 3 are used to denote the four cases. If the sign indication is 2 or 3, the sign bit does not have to be coded because it is either associated with a zero difference or available from the corresponding base layer data. If the sign indication is 0 or 1 a sign bit code is required once per difference value, i.e. not every bit-plane of the difference value. Therefore, a sign bit is put immediately after the most significant bit of the difference.

Optimal Reconstruction of the DCT Coefficients

In an alternative embodiment of the present invention, even though N enhancement bitstream layers or planes may have been generated, only M, wherein M $\leq$ N enhancement layer bits are available for reconstruction of the DCT coefficients due to the channel capacity, and other constraints such as congestion among others, the decoder 80 of Fig. 1 may receive no enhancement difference or only a partial enhancement difference. In such a case, the optimal reconstruction of the DCT coefficients is capable of proceeding along the following method:

If decoded difference = 0, the reconstruction point is the same as that in base layer, otherwise, the reconstructed difference = decoded difference + ¼ *(1<<decoded_bit_plane) and the reconstruction point = reference point + reconstructed difference * Q_enh +Q_enh/2.

In the present embodiment, referring to Figs. 3C and 3D, the optimal reconstruction point is not the lower boundary of a quantization bin. The above method specifies how to obtain the optimal reconstruction point in cases where the difference is quantized and received partially, i.e. not all of the enhancement layers generated are either transmitted or received as shown in Fig. 1. wherein $M \leq N$.

What is claimed is:

1. A video encoding method for adapting a video input to a bandwidth of a transmission channel of a network, the method comprising the steps of:

determining number N of enhancement layer bitstreams capable of being adapted to said bandwidth of said transmission channel of said network;

encoding a base layer bitstream from said video input;

encoding N number of enhancement layer bitstreams from said video input based on the base layer bitstream, wherein the N enhancement layer bitstreams complements the base layer bitstream; and

providing the base layer bitstream and N enhancement layer bitstreams to said network.

2. The video encoding method according to claim 1, wherein the determining step includes negotiating with intermediate devices on said network.

3. The video encoding method according to claim 2, wherein negotiating includes determining destination resources.

4. The video encoding method according to claim 1, wherein the step of encoding the base layer bitstreams is performed by a MPEG-1 encoding method.

5. The video encoding method according to claim 1, wherein the step of encoding the base layer bitstreams is performed by a MPEG-2 encoding method.

6. The video encoding method according to claim 1, wherein the step of encoding the base layer bitstreams is performed by a MPEG-4 encoding method.

7. The video encoding method according to claim 1, wherein the step of encoding the base layer bitstreams is performed by a Discrete Cosine Transform (DCT) method.

8. The video encoding method according to claim 7, wherein after encoding the base layer bitstreams by a Discrete Cosine Transform (DCT) method a DCT coefficient is quantized.

9. The video encoding method according to claim 1, wherein the enhancement layer bitstreams are based on the difference of an original base layer DCT coefficient and a corresponding base layer quantized DCT coefficient.

10. The video encoding method according to claim 1, wherein the base layer bitstream and the N enhancement layer provide to the network are multiplexed.

11. A video decoding method for adapting a video input to a bandwidth of a transmission channel of a network, the method comprising the steps of:

determining number M of enhancement layer bitstreams of said video input capable of being received from said transmission channel of said network;

decoding a base layer bitstream from received video input;

decoding M number of enhancement layer bitstreams from the received video input based on the base layer bitstream, wherein the M received enhancement layer bitstreams complements the base layer bitstream; and

reconstructing the base layer bitstream and N enhancement layer bitstreams.

12. The video decoding method according to claim 11, wherein the determining step includes negotiating with intermediate devices on said network.

13. The video decoding method according to claim 12, wherein negotiating includes determining destination resources.

14. The video decoding method according to claim 11, wherein the step of decoding the base layer bitstreams is performed by a MPEG-1 decoding method.

15. The video decoding method according to claim 11, wherein the step of decoding the base layer bitstreams is performed by a MPEG-2 decoding method.

16. The video decoding method according to claim 11, wherein the step of decoding the base layer bitstreams is performed by a MPEG-4 decoding method.

17. The video decoding method according to claim 11, wherein the step of decoding the base layer bitstreams is performed by a Discrete Cosine Transform (DCT) method.

18. The video decoding method according to claim 17, wherein after decoding the base layer bitstreams by a Discrete Cosine Transform (DCT) method a DCT coefficient is unquantized.

19. The video decoding method according to claim 11, wherein coding of the enhancement layer bitstreams are based on the difference of an original base layer DCT coefficient and a corresponding base layer quantized DCT coefficient.

20. The video decoding method according to claim 11, wherein the base layer bitstream and the M enhancement layers to be reconstructed are de-multiplexed.

21. A video decoding method for adapting a video input to a bandwidth of a receiving apparatus, the method comprising the steps of:
demultiplexing a base layer bitstream and at least one of a plurality of enhancement layer bitstreams received from a network;
decoding the base layer bitstream;
decoding at least one of the plurality of enhancement layer bitstreams based on generated base layer bitstream, wherein the at least one of the plurality of enhancement layer bitstreams enhances the base layer bitstream; and
reconstructing a video output.

22. A video encoding method for encoding enhancement layers based on a base layer bitstream encoded from a video input, the video encoding method comprising the steps of:

taking a difference between an original DCT coefficient and a reference point; and

dividing the difference between the original DCT coefficient and the reference point into N bit-planes.

23. The video encoding method according to claim 22, wherein RUN and EOP symbols represents the N bit-planes of a DCT block.

24. The video encoding method according to claim 23, wherein the RUN and EOP symbols are encoded.

25. The video encoding method according to claim 24, wherein a sign bit is encoded if the DCT difference is equal to zero or the sign of the DCT difference is the same as the corresponding base layer bitstream data.

26. A video decoding method for reconstructing DCT coefficients M enhancement layers of N enhancement layers have been received, wherein $M \leq N$, comprising:

means for taking a reconstruction difference as a decoded difference and a portion of a decoded bit-plane;

means for taking a reconstruction point as a reference point and a reconstructed difference; and

determining an optimal reconstruction point.

27. A method of coding motion vectors of a plurality of macroblocks, the method comprising the steps of:

determining an average motion vector from N motion vectors for N macroblocks;

utilizing the determined average motion vector as the motion vector for the N macroblocks; and

encoding 1/N motion vectors in a base layer bitstream.

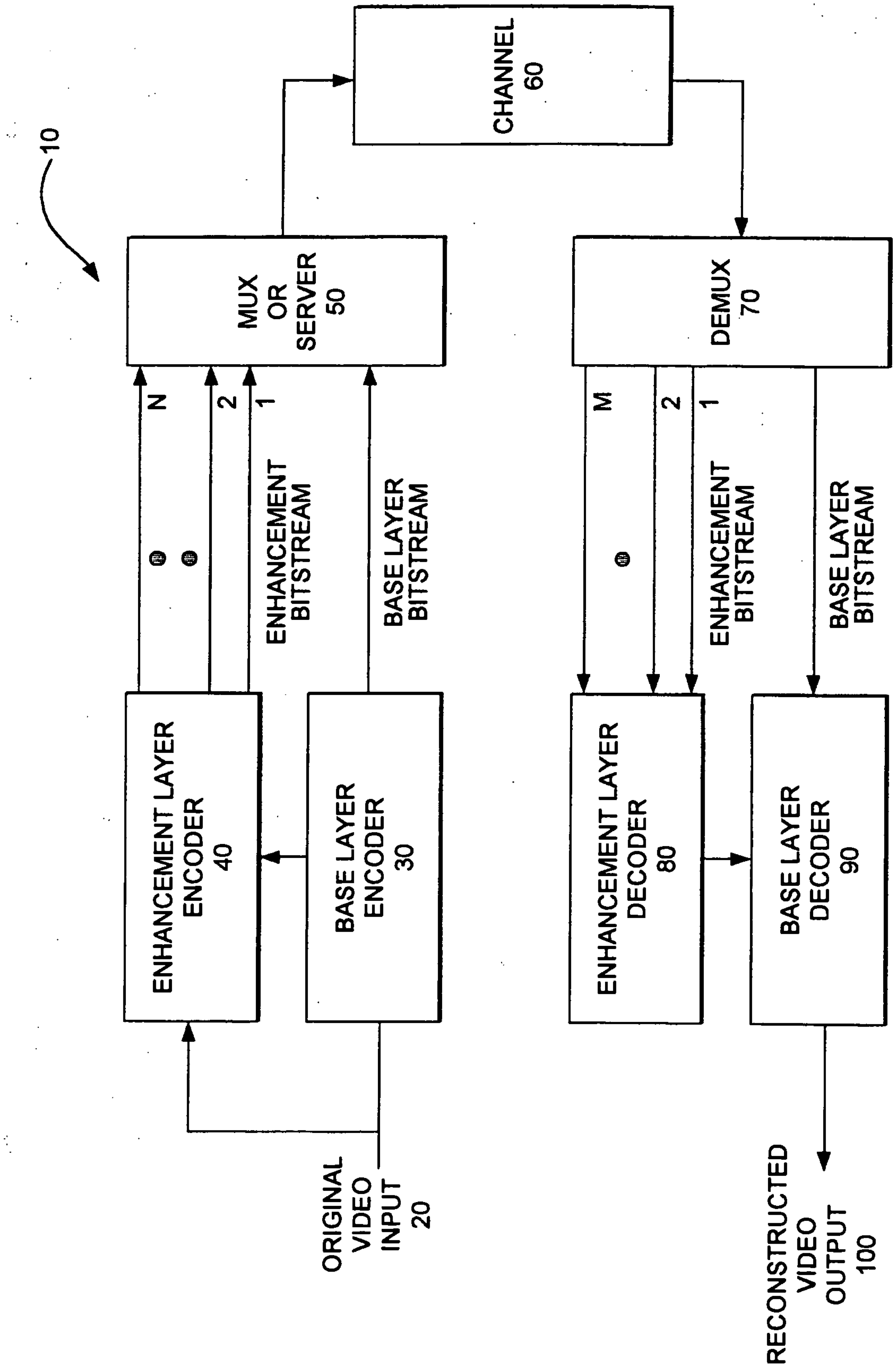

10
CHANNEL 60
MUX OR SERVER 50
DEMUX 70
N
2
1
ENHANCEMENT BITSTREAM
BASE LAYER BITSTREAM
M
2
1
ENHANCEMENT BITSTREAM
BASE LAYER BITSTREAM
ENHANCEMENT LAYER ENCODER 40
BASE LAYER ENCODER 30
ENHANCEMENT LAYER DECODER 80
BASE LAYER DECODER 90
ORIGINAL VIDEO INPUT 20
RECONSTRUCTED VIDEO OUTPUT 100

FIG. 1

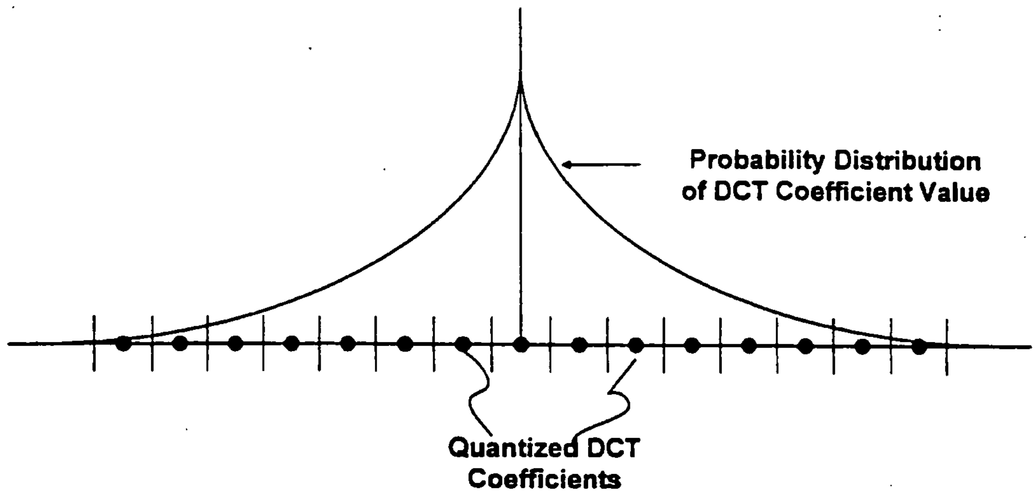
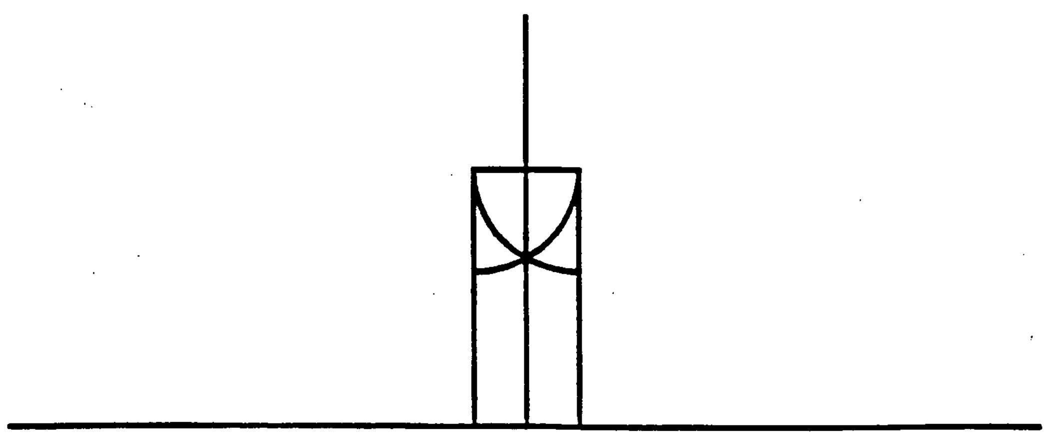


FIG. 2A



Probability Distribution of DCT Coefficient Residue

FIG. 2B

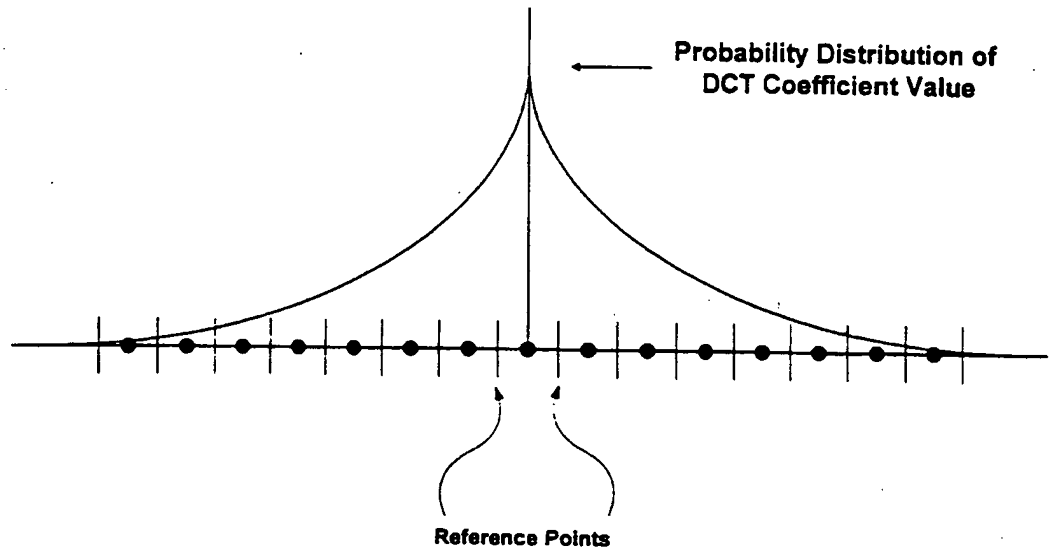

Probability Distribution of
DCT Coefficient Value
Reference Points

FIG. 3A

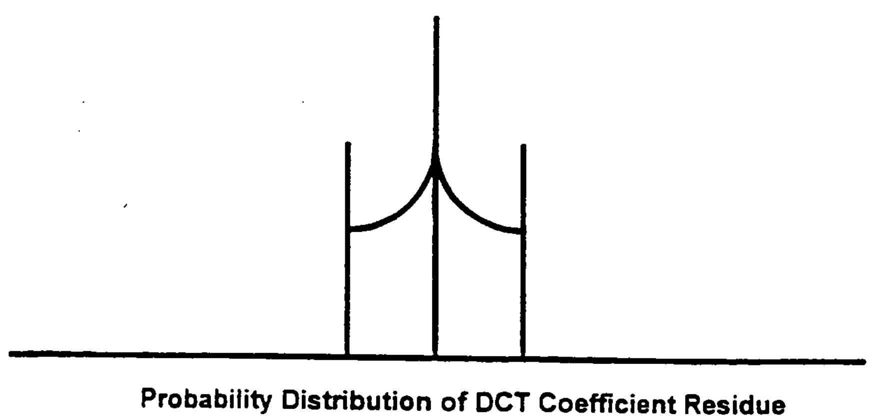

Probability Distribution of DCT Coefficient Residue

FIG. 3B

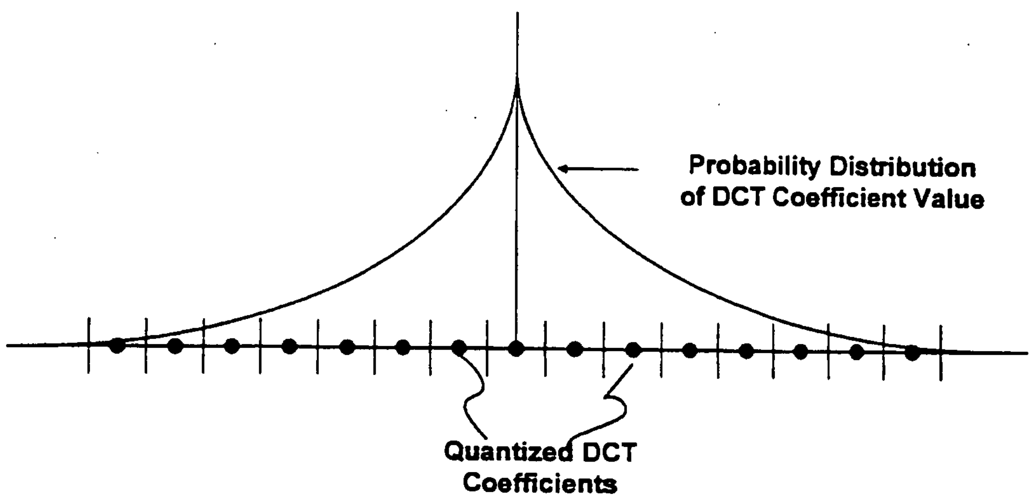

Probability Distribution
of DCT Coefficient Value
Quantized DCT
Coefficients

FIG. 3C

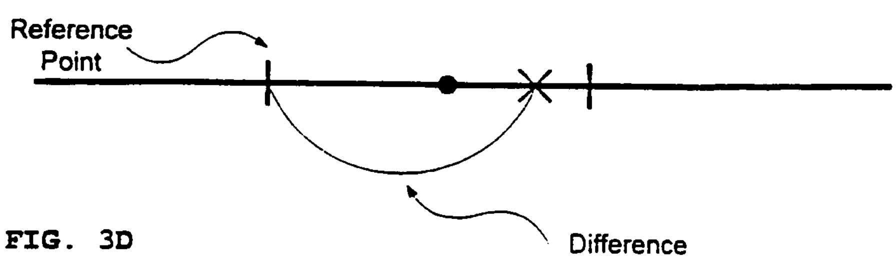

Reference
Point
Difference

FIG. 3D

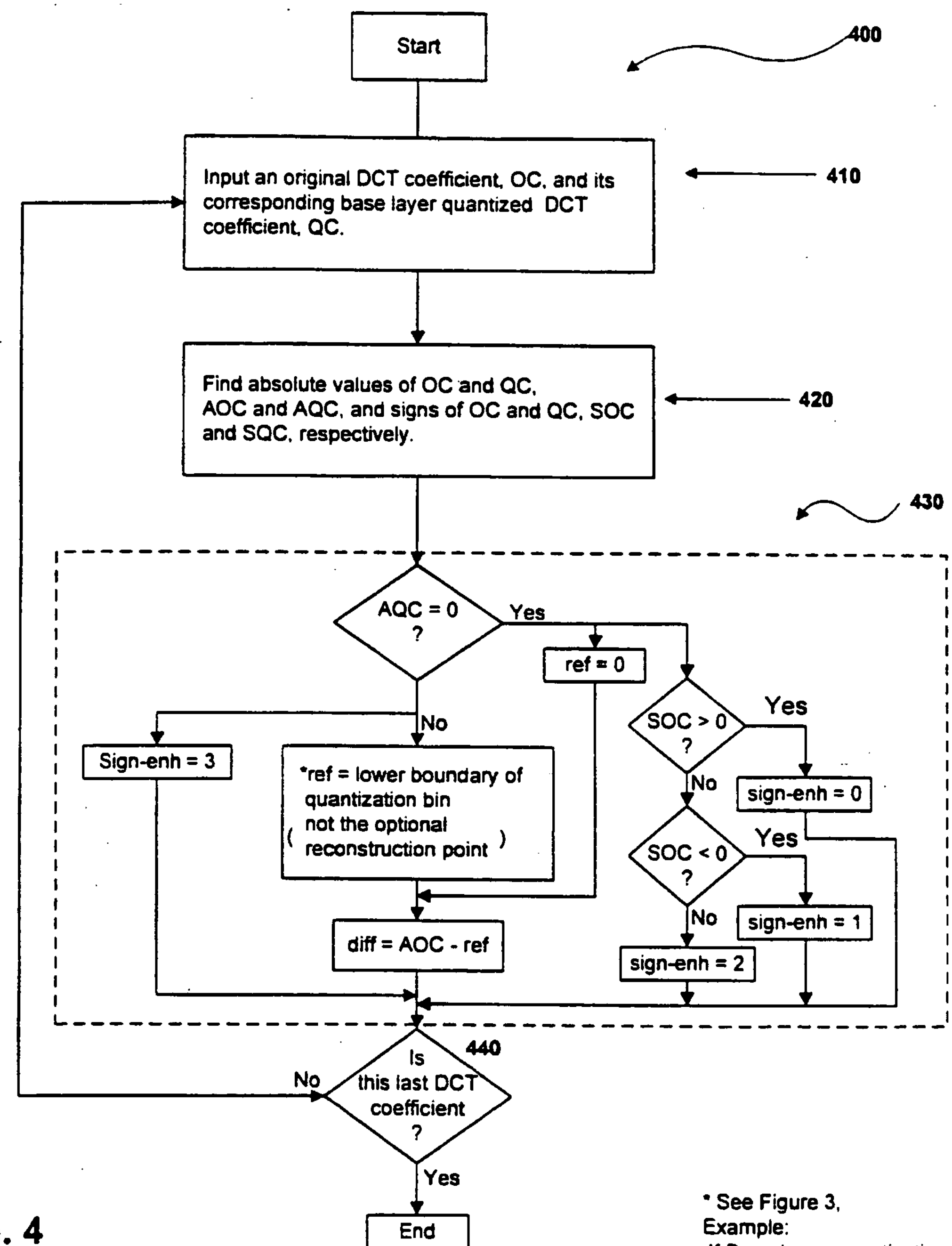


FIG. 4

* See Figure 3,
Example:
If Base Layer quantization is
AQC = AOC / (2*Q)
lower boundary is AQC * (2*Q)
optimal point is AQC*(2*Q) + Q

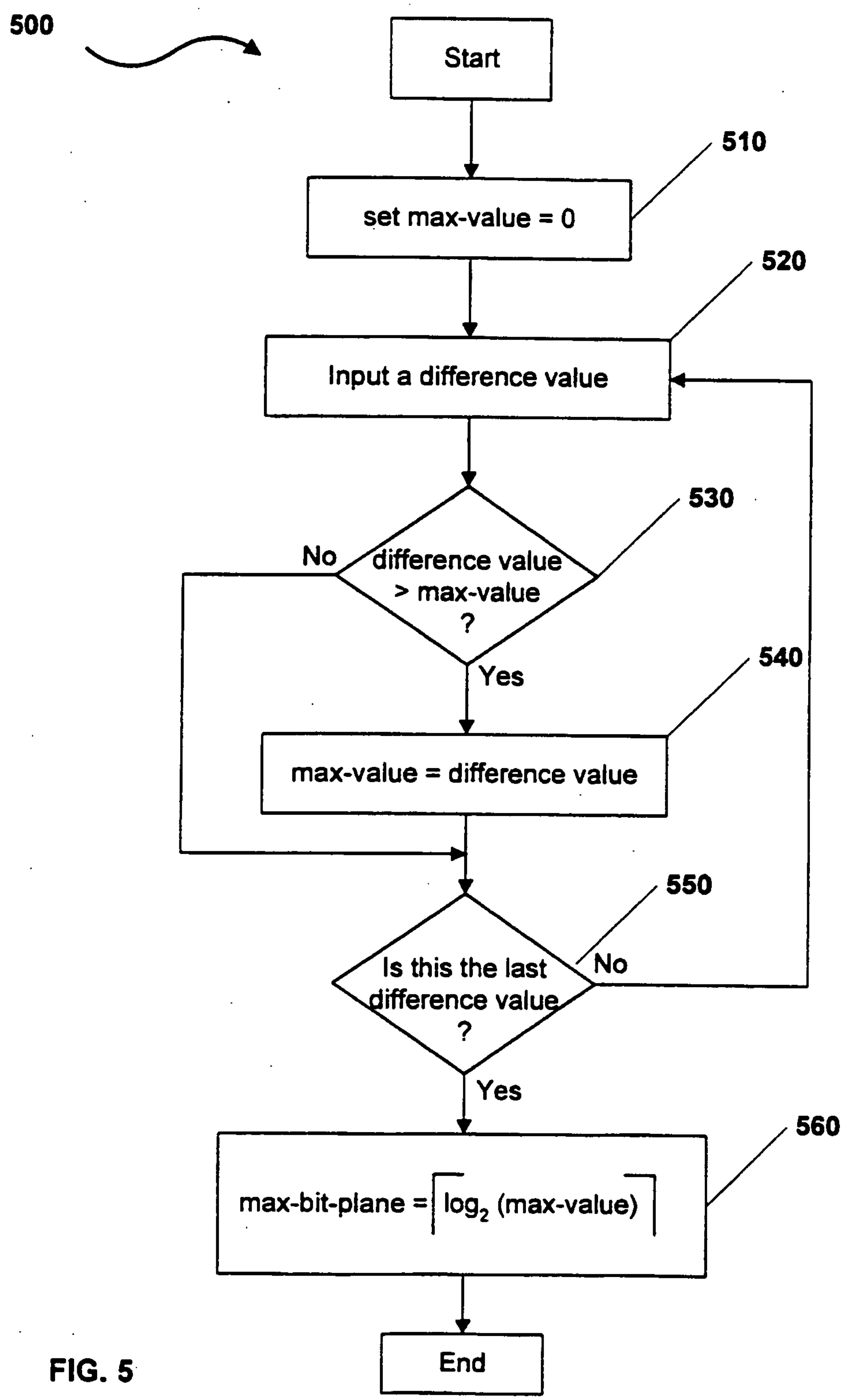

500
Start
510
set max-value = 0
520
Input a difference value
530
No
difference value
> max-value
?
Yes
540
max-value = difference value
550
Is this the last
difference value
?
No
Yes
560
max-bit-plane = ⌈log2 (max-value)⌉
End

FIG. 5

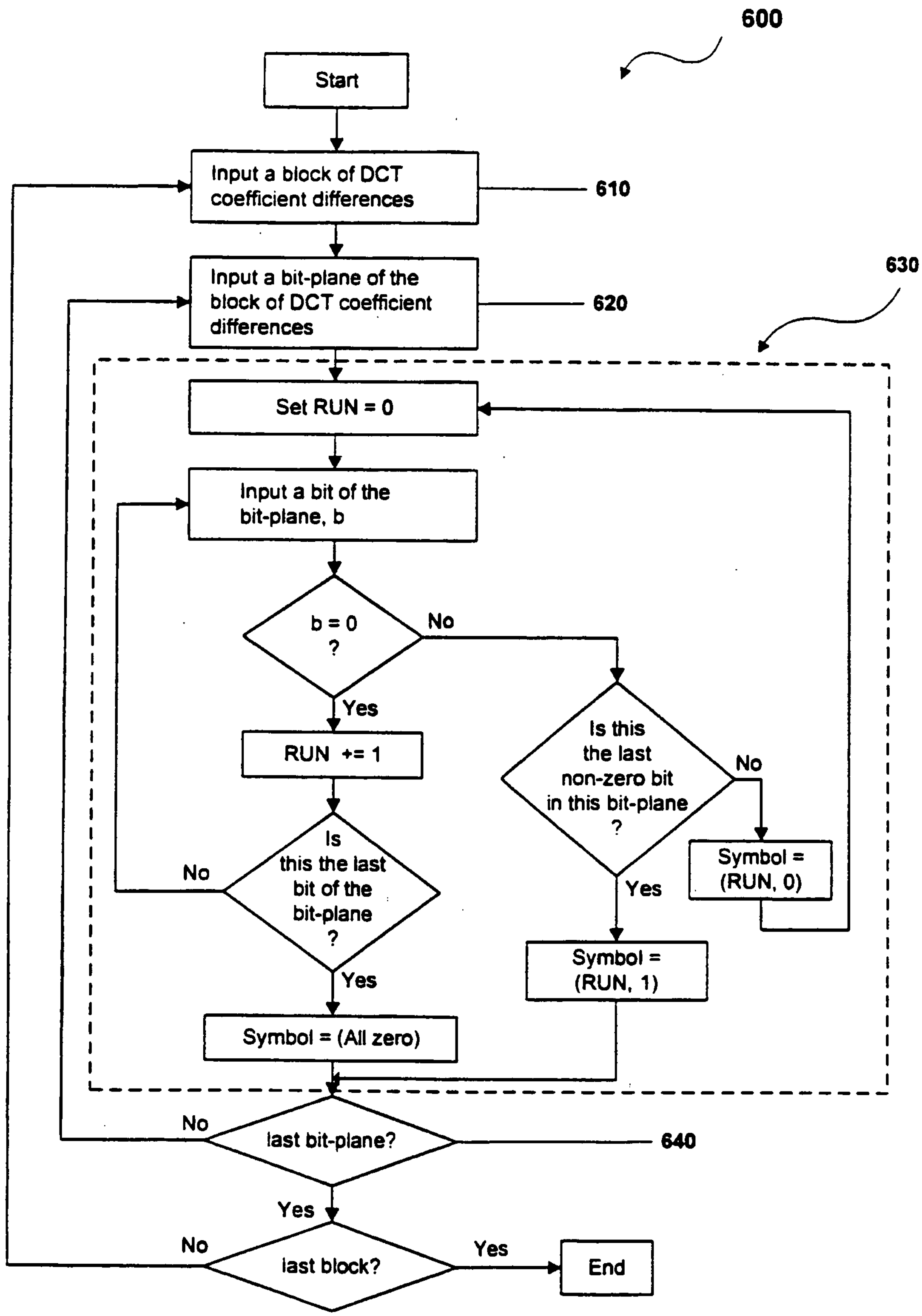

600
Start
Input a block of DCT coefficient differences
610
Input a bit-plane of the block of DCT coefficient differences
620
630
Set RUN = 0
Input a bit of the bit-plane, b
b = 0 ?
No
Yes
RUN += 1
Is this the last non-zero bit in this bit-plane ?
No
Symbol = (RUN, 0)
Is this the last bit of the bit-plane ?
No
Yes
Yes
Symbol = (RUN, 1)
Symbol = (All zero)
No
last bit-plane?
640
Yes
No
last block?
Yes
End

FIG. 6

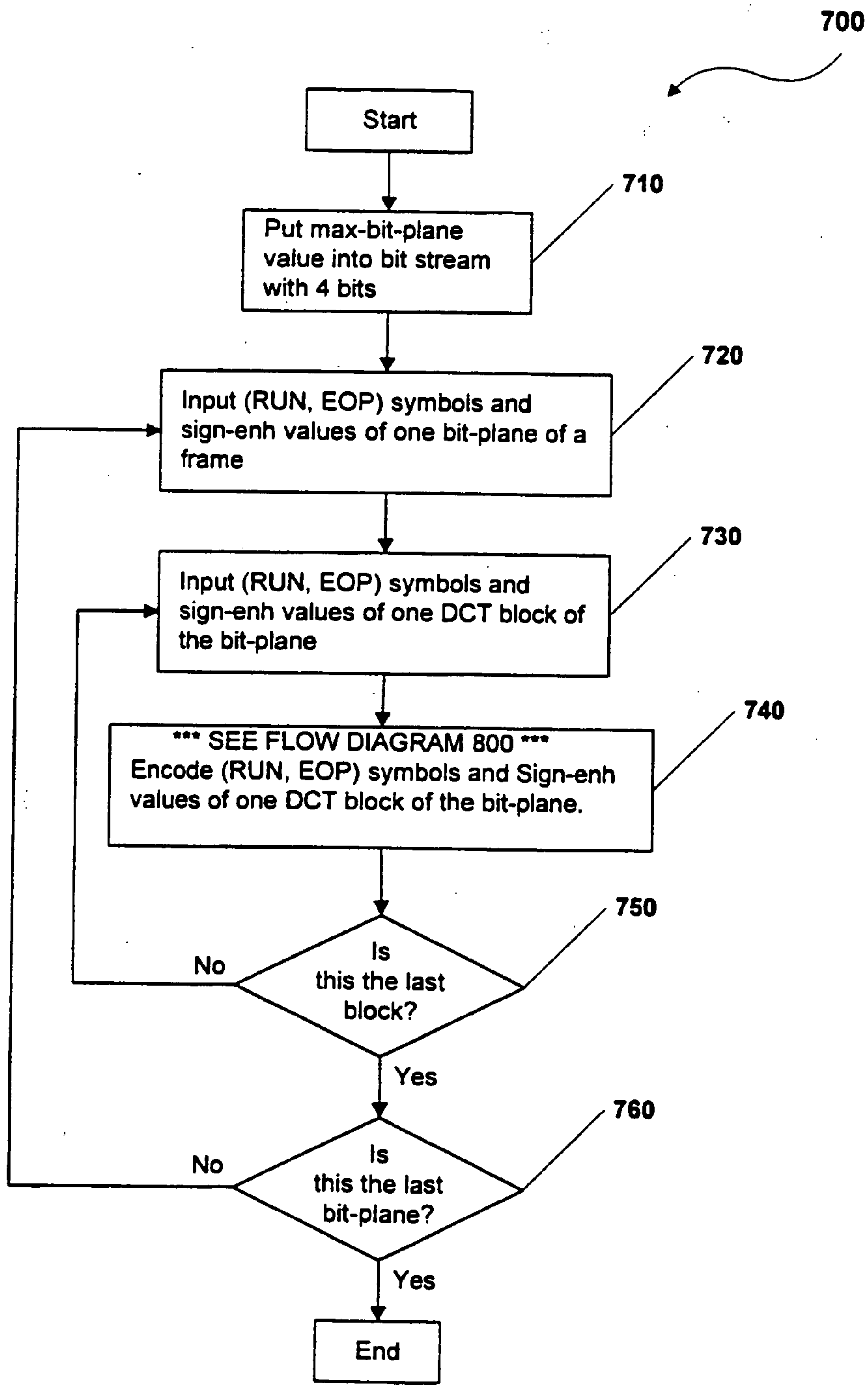

700
Start
710
Put max-bit-plane value into bit stream with 4 bits
720
Input (RUN, EOP) symbols and sign-enh values of one bit-plane of a frame
730
Input (RUN, EOP) symbols and sign-enh values of one DCT block of the bit-plane
740
*** SEE FLOW DIAGRAM 800 ***
Encode (RUN, EOP) symbols and Sign-enh values of one DCT block of the bit-plane.
750
Is this the last block?
No
Yes
760
Is this the last bit-plane?
No
Yes
End

FIG. 7

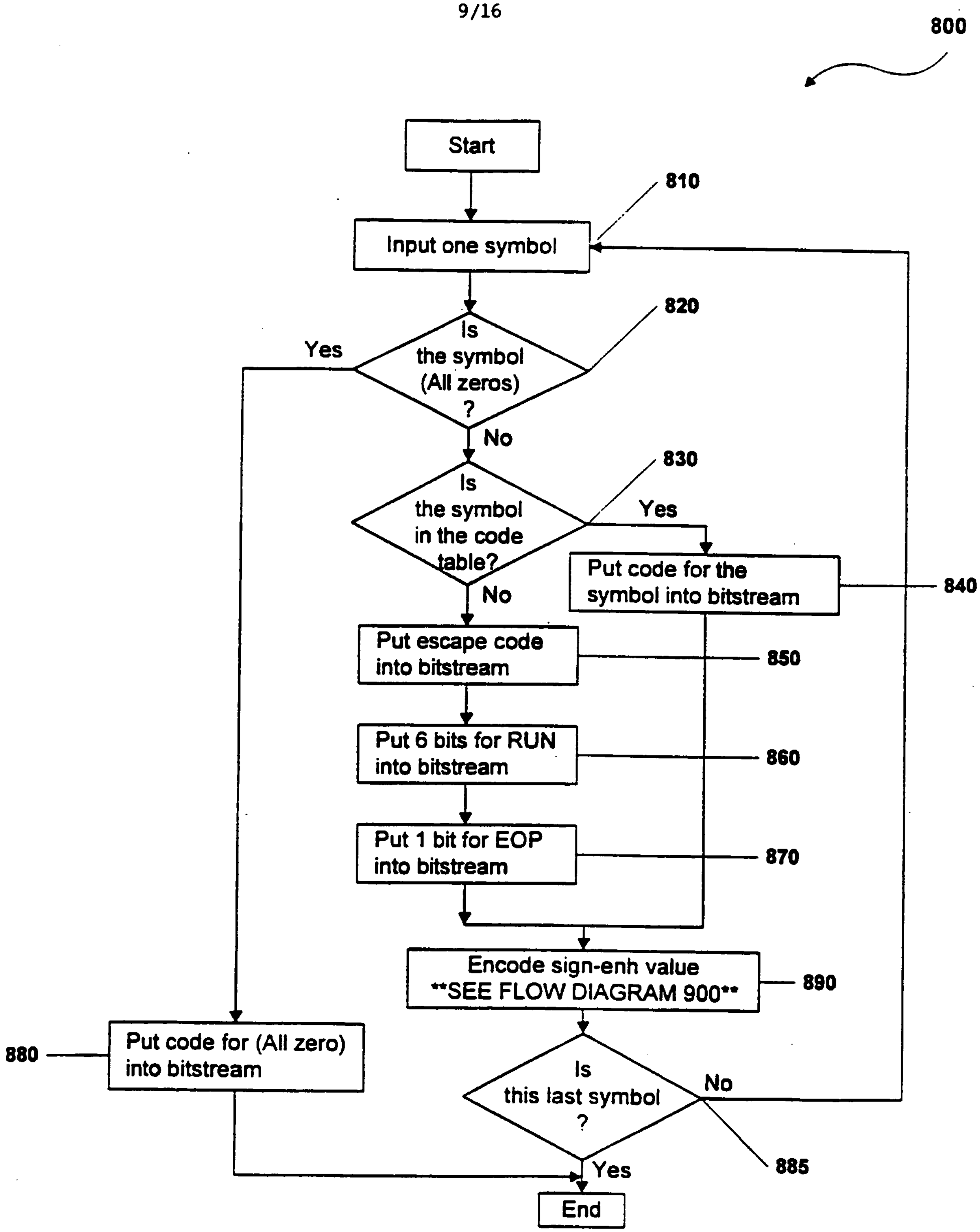

800
Start
810
Input one symbol
820
Is the symbol (All zeros) ?
Yes
No
830
Is the symbol in the code table?
Yes
No
Put code for the symbol into bitstream
840
Put escape code into bitstream
850
Put 6 bits for RUN into bitstream
860
Put 1 bit for EOP into bitstream
870
Encode sign-enh value **SEE FLOW DIAGRAM 900**
890
880
Put code for (All zero) into bitstream
Is this last symbol ?
No
Yes
885
End

FIG. 8

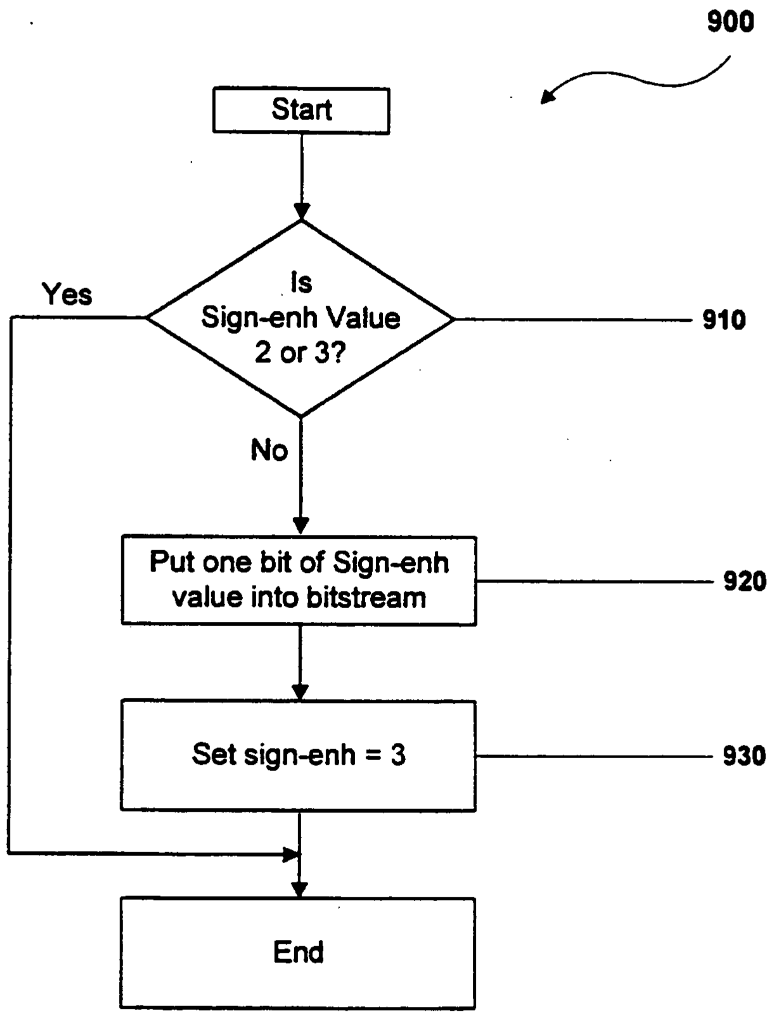

900
Start
Is
Sign-enh Value
2 or 3?
Yes
No
910
Put one bit of Sign-enh
value into bitstream
920
Set sign-enh = 3
930
End

FIG. 9

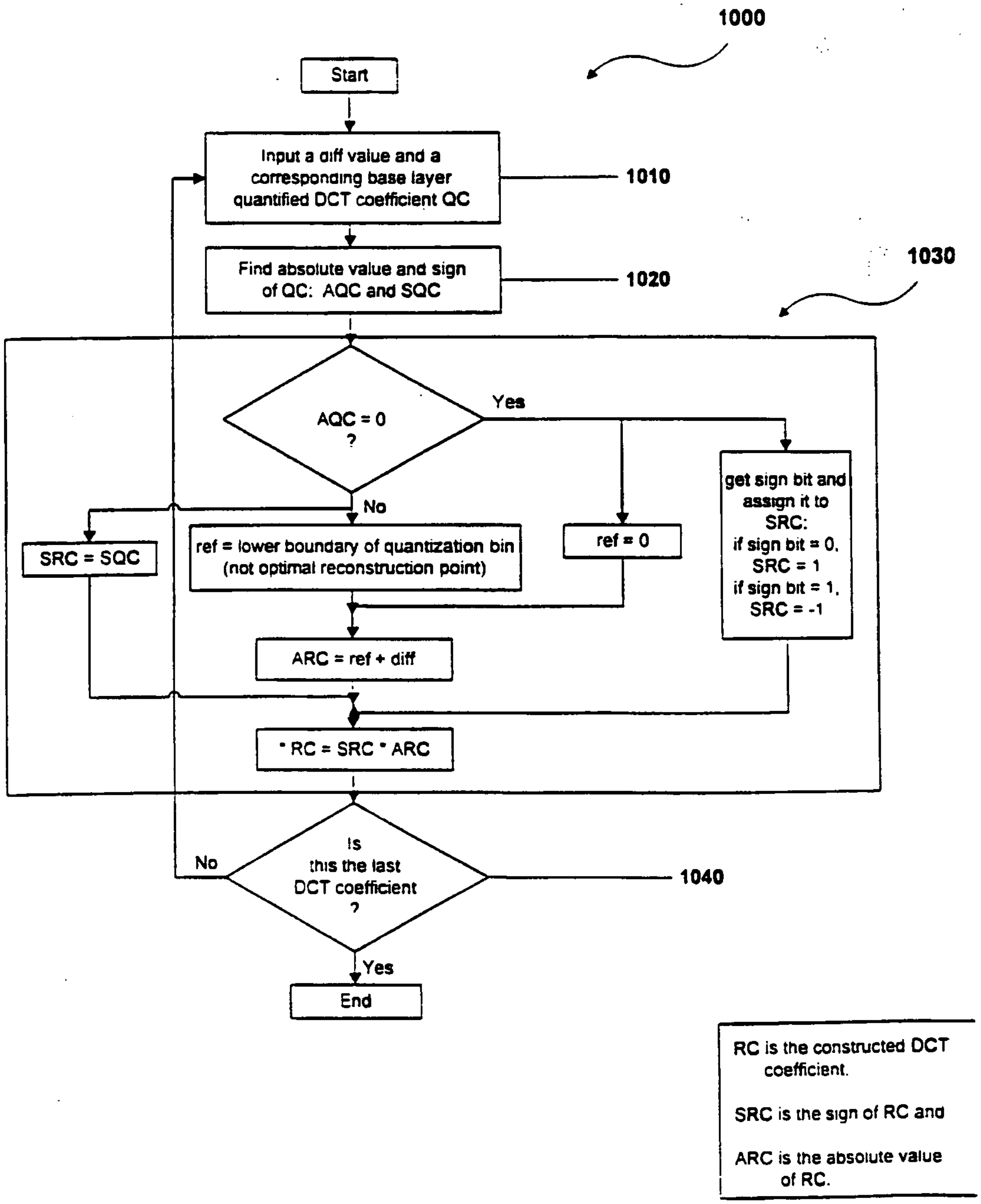

1000
Start
Input a diff value and a corresponding base layer quantified DCT coefficient QC
1010
Find absolute value and sign of QC: AQC and SQC
1020
1030
AQC = 0 ?
Yes
No
SRC = SQC
ref = lower boundary of quantization bin (not optimal reconstruction point)
ref = 0
get sign bit and assign it to SRC: if sign bit = 0, SRC = 1 if sign bit = 1, SRC = -1
ARC = ref + diff
RC = SRC * ARC
Is this the last DCT coefficient ?
No
Yes
1040
End
RC is the constructed DCT coefficient.
SRC is the sign of RC and
ARC is the absolute value of RC.

FIG. 10

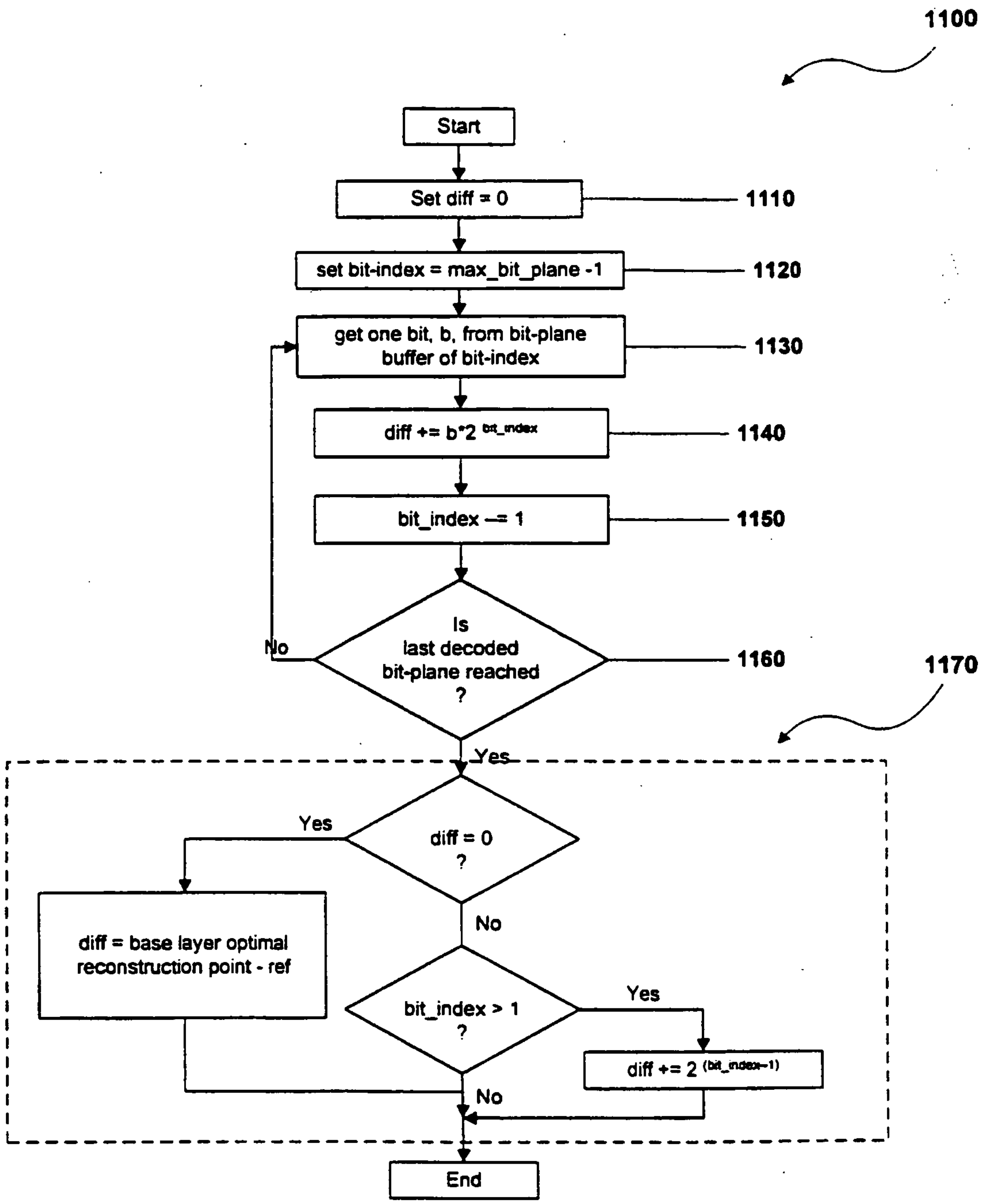

1100
Start
Set diff = 0
1110
set bit-index = max_bit_plane -1
1120
get one bit, b, from bit-plane buffer of bit-index
1130
diff += b*2 bit_index
1140
bit_index -= 1
1150
Is last decoded bit-plane reached ?
1160
No
Yes
1170
diff = 0 ?
Yes
diff = base layer optimal reconstruction point - ref
No
bit_index > 1 ?
Yes
diff += 2 (bit_index-1)
No
End

FIG. 11

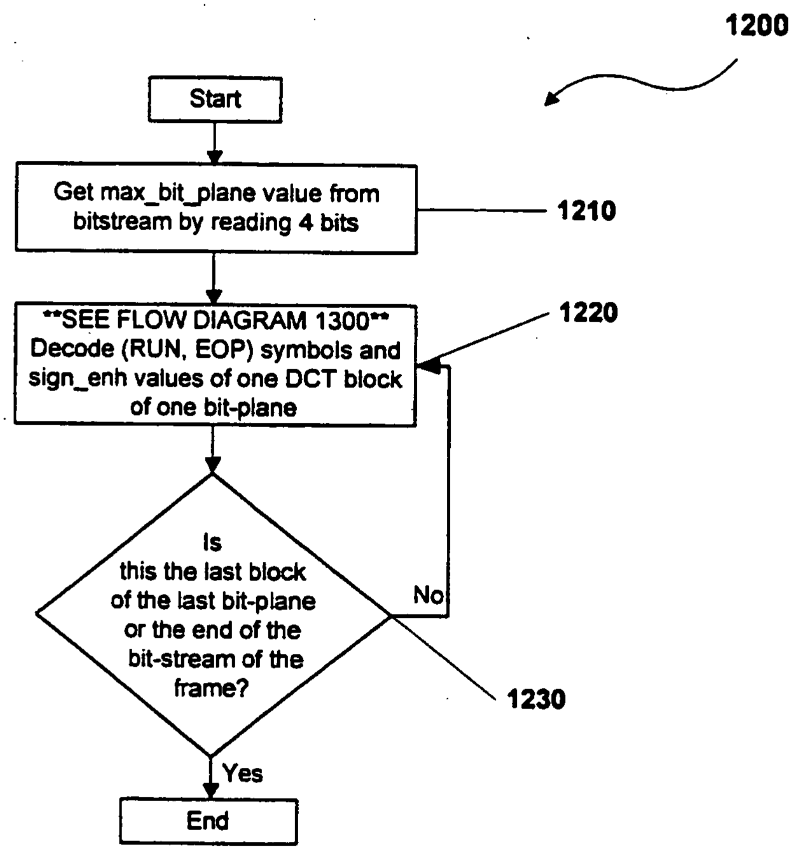

1200
Start
Get max_bit_plane value from bitstream by reading 4 bits
1210
SEE FLOW DIAGRAM 1300
Decode (RUN, EOP) symbols and sign_enh values of one DCT block of one bit-plane
1220
Is this the last block of the last bit-plane or the end of the bit-stream of the frame?
No
1230
Yes
End

FIG. 12

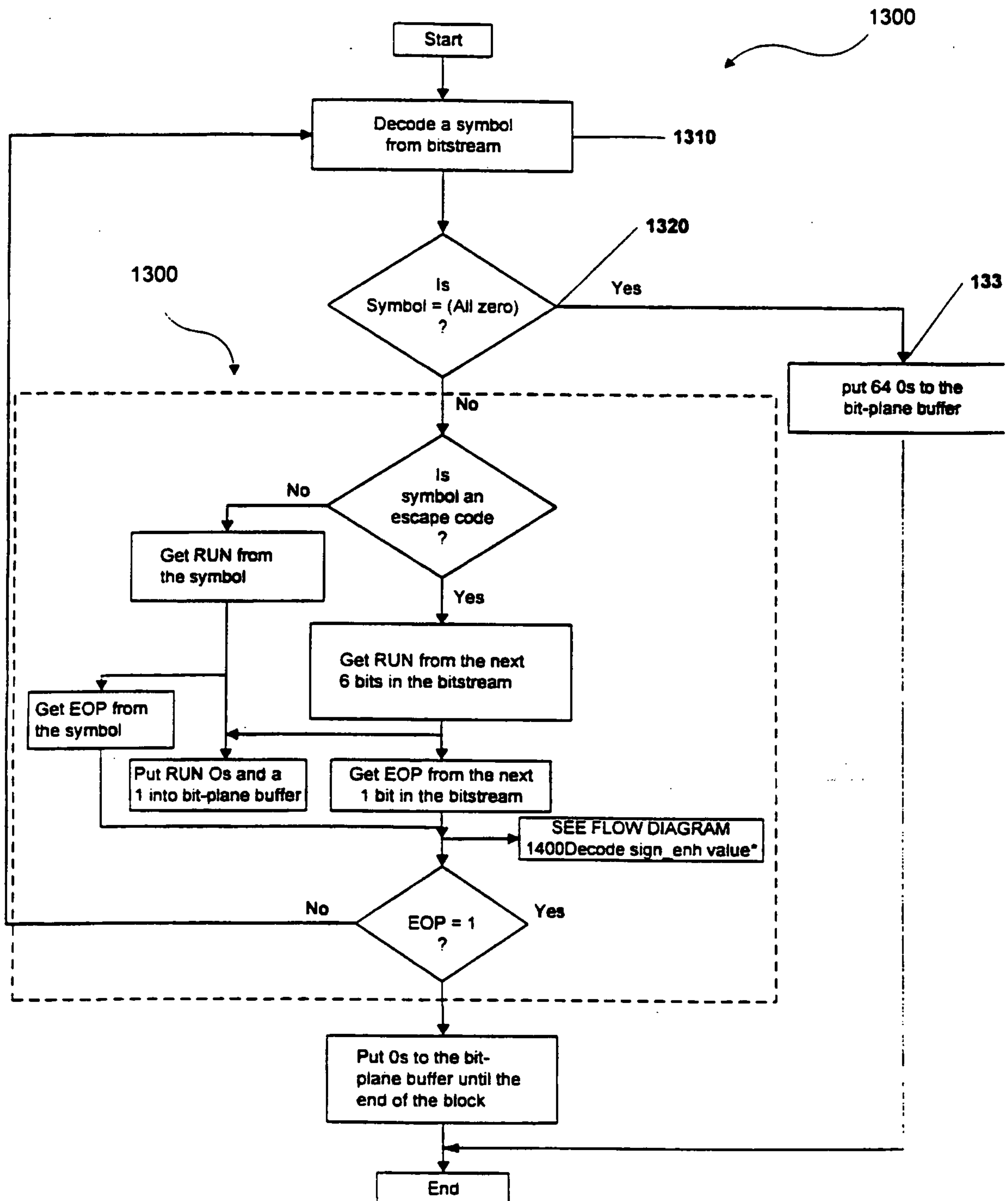

1300
Start
Decode a symbol from bitstream
1310
1320
Is Symbol = (All zero) ?
Yes
133
put 64 0s to the bit-plane buffer
1300
No
Is symbol an escape code ?
No
Yes
Get RUN from the symbol
Get RUN from the next 6 bits in the bitstream
Get EOP from the symbol
Put RUN 0s and a 1 into bit-plane buffer
Get EOP from the next 1 bit in the bitstream
SEE FLOW DIAGRAM 1400Decode sign_enh value*
No
EOP = 1 ?
Yes
Put 0s to the bit-plane buffer until the end of the block
End

FIG. 13

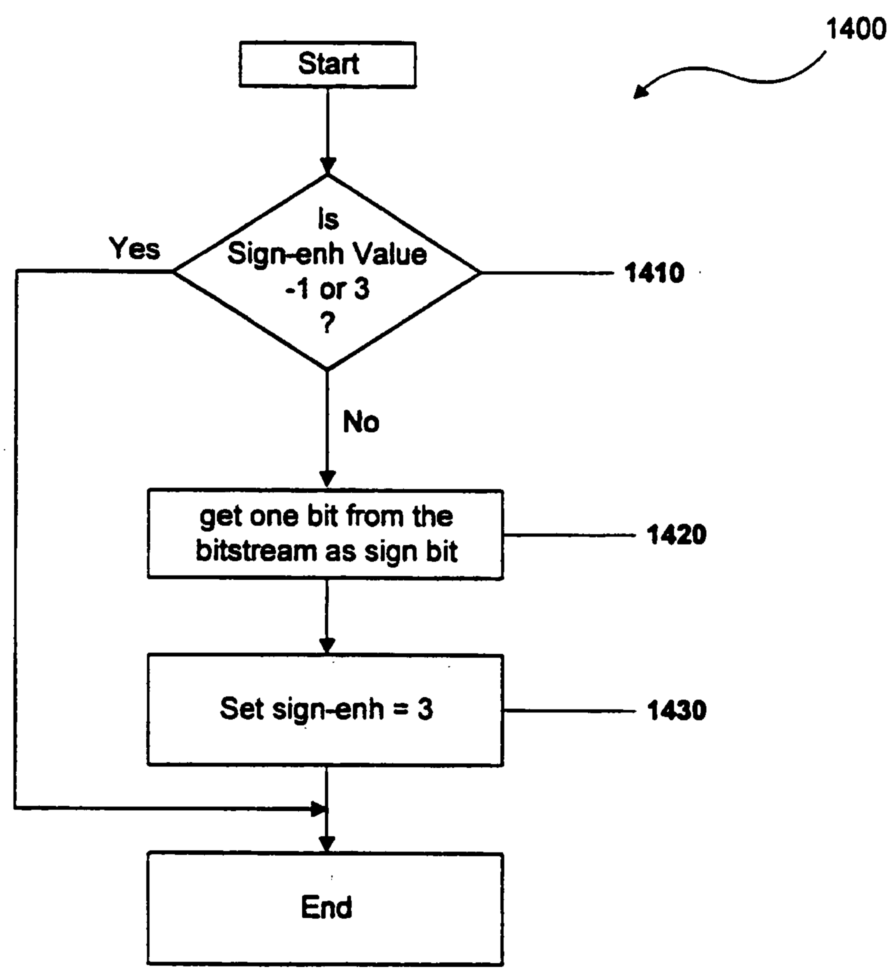

1400
Start
Is
Sign-enh Value
-1 or 3
?
Yes
1410
No
get one bit from the
bitstream as sign bit
1420
Set sign-enh = 3
1430
End

FIG. 14

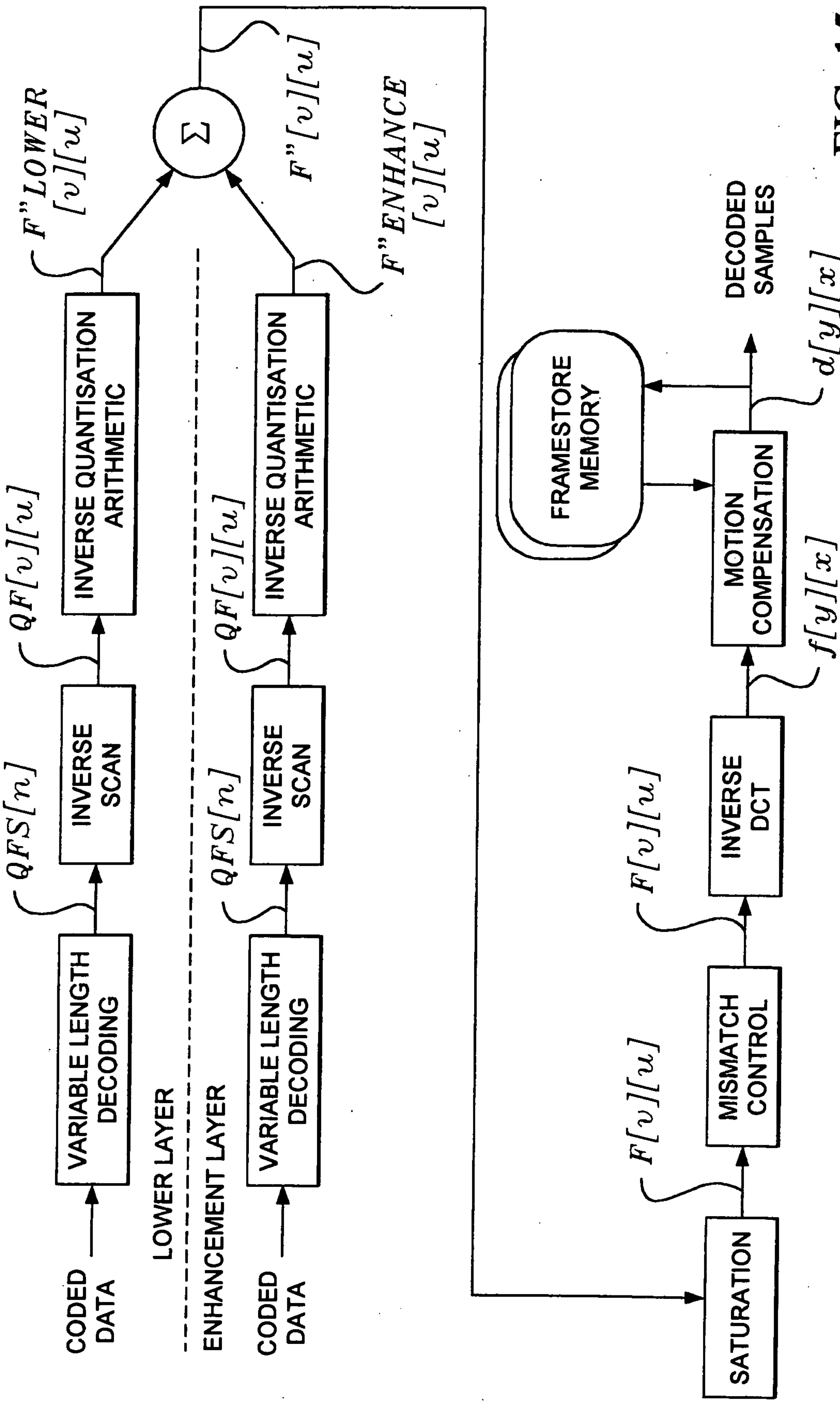


FIG. 15
PRIOR ART

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04L 9/00	A2	(11) International Publication Number: WO 00/59152 (43) International Publication Date: 5 October 2000 (05.10.00)

(21) International Application Number: PCT/US00/04983

(22) International Filing Date: 25 February 2000 (25.02.00)

(30) Priority Data:

60/126,614	27 March 1999 (27.03.99)	US
09/290,363	12 April 1999 (12.04.99)	US
09/482,928	13 January 2000 (13.01.00)	US

(71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052 (US).

(72) Inventors: BLINN, Arnold, N.; 9401 NE 27th Street, Bellevue, WA 98004 (US). JONES, Thomas, C.; 23617 NE 6th Street, Redmond, WA 98053-3618 (US).

(74) Agents: ROCCI, Steven, J. et al.; Woodcock Washburn Kurtz Mackiewicz & Norris LLP, 46th floor, One Liberty Place, Philadelphia, PA 19103 (US).

(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, 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).

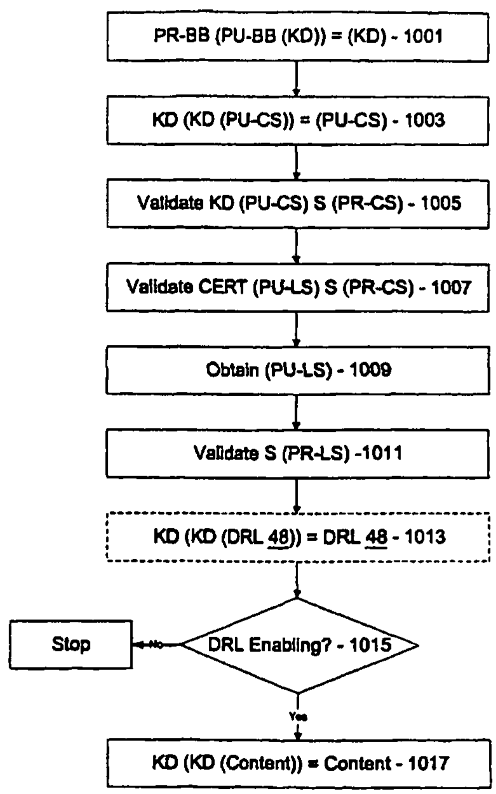
Published

Without international search report and to be republished upon receipt of that report.

(54) Title: METHOD FOR INTERDEPENDENTLY VALIDATING A DIGITAL CONTENT PACKAGE AND A CORRESPONDING DIGITAL LICENSE

(57) Abstract

A method is disclosed for a device to interdependently validate a digital content package having a piece of digital content in an encrypted form, and a corresponding digital license for rendering the digital content. A first key is derived from a source available to the device, and a first digital signature is obtained from the digital content package. The first key is applied to the first digital signature to validate the first digital signature and the digital content package. A second key is derived based on the first digital signature, and a second digital signature is obtained from the license. The second key is applied to the second digital signature to validate the second digital signature and the license.



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 FOR INTERDEPENDENTLY VALIDATING A DIGITAL CONTENT PACKAGE AND A CORRESPONDING DIGITAL LICENSE

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. Patent Application No. 09/290,363, filed April 12, 1999 and entitled "ENFORCEMENT ARCHITECTURE AND METHOD FOR DIGITAL RIGHTS MANAGEMENT", and claims the benefit of U.S. Provisional Application No. 60/21,614, filed March 27, 1999 and entitled "ENFORCEMENT ARCHITECTURE AND METHOD FOR DIGITAL RIGHTS MANAGEMENT", both of which are hereby incorporated by reference.

TECHNICAL FIELD

The present invention relates to an architecture for enforcing rights in digital content. More specifically, the present invention relates to such an enforcement architecture that allows access to encrypted digital content only in accordance with parameters specified by license rights acquired by a user of the digital content.

BACKGROUND OF THE INVENTION

Digital rights management and enforcement is highly desirable in connection with digital content such as digital audio, digital video, digital text, digital data, digital multimedia, etc., where such digital content is to be distributed to users. Typical modes of distribution include tangible devices such as a magnetic (floppy) disk, a magnetic tape, an optical (compact) disk (CD), etc., and intangible media such as an electronic bulletin board, an electronic network, the Internet, etc. Upon being received by the user, such user renders or 'plays' the digital content with the aid of an appropriate rendering device such as a media player on a personal computer or the like.

Typically, a content owner or rights-owner, such as an author, a publisher, a broadcaster, etc. (hereinafter "content owner"), wishes to distribute such digital content to a user or recipient in exchange for a license fee or some other consideration. Such content owner, given the choice, would likely wish to restrict what the user can do with such distributed digital content. For example, the content owner would like to restrict the user from copying and re-distributing such content to a second user, at least in a manner that denies the content owner a license fee from such second user.

In addition, the content owner may wish to provide the user with the flexibility to purchase different types of use licenses at different license fees, while at the same time holding the user to the terms of whatever type of license is in fact purchased. For example, the content owner may wish to allow distributed digital content to be played only a limited number of times, only for a certain total time, only on a certain type of machine, only on a certain type of media player, only by a certain type of user, etc.

However, after distribution has occurred, such content owner has very little if any control over the digital content. This is especially problematic in view of the fact that practically every new or recent personal computer includes the software and hardware necessary to make an exact digital copy of such digital content, and to download such exact digital copy to a write-able magnetic or optical disk, or to send such exact digital copy over a network such as the Internet to any destination.

Of course, as part of the legitimate transaction where the license fee was obtained, the content owner may require the user of the digital content to promise not to re-distribute such digital content. However, such a promise is easily made and easily broken. A content owner may attempt to prevent such re-distribution through any of several known security devices, usually involving encryption and decryption. However, there is likely very little that prevents a mildly determined user from decrypting encrypted digital content, saving such digital content in an un-encrypted form, and then re-distributing same.

A need exists, then, for providing an enforcement architecture and method that allows the controlled rendering or playing of arbitrary forms of digital content, where such control is flexible and definable by the content owner of such digital content. A need also exists for providing a controlled rendering environment on a computing device such as a personal computer, where the rendering environment includes at least a portion of such enforcement architecture. Such controlled rendering environment allows that the digital content will only be rendered as specified by the content owner, even though the digital content is to be rendered on a computing device which is not under the control of the content owner.

Further, a need exists for a trusted component running on the computing device, where the trusted component enforces the rights of the content owner on such computing device in connection with a piece of digital content, even against attempts by the user of such computing device to access such digital content in ways not permitted by the content owner. As but one example, such a trusted software component prevents a user of the computing device from making a copy of such digital content, except as otherwise allowed for by the content owner thereof.

SUMMARY OF THE INVENTION

The aforementioned needs are satisfied at least in part by an enforcement architecture and method for digital rights management, where the architecture and method enforce rights in protected (secure) digital content available on a medium such as the Internet, an optical disk, etc. For purposes of making content available, the architecture includes a content server from which the digital content is accessible over the Internet or the like in an encrypted form. The content server may also supply the encrypted digital content for recording on an optical disk or the like, wherein the encrypted digital content may be distributed on the optical disk itself. At the content server, the digital content is encrypted using an encryption key, and public / private key techniques are employed to bind the digital content with a digital license at the user's computing device or client machine.

When a user attempts to render the digital content on a computing device, the rendering application invokes a Digital Rights Management (DRM) system on such user's computing device. If the user is attempting to render the digital content for the first time, the DRM system either directs the user to a license server to obtain a license to render such digital content in the manner sought, or transparently obtains such license from such license server without any action necessary on the part of the user. The license includes:

- a decryption key (KD) that decrypts the encrypted digital content;
- a description of the rights (play, copy, etc.) conferred by the license and related conditions (begin date, expiration date, number of plays, etc.), where such description is in a digitally readable form; and
- a digital signature that ensures the integrity of the license.

The user cannot decrypt and render the encrypted digital content without obtaining such a license from the license server. The obtained license is stored in a license store in the user's computing device.

Importantly, the license server only issues a license to a DRM system that is 'trusted' (i.e., that can authenticate itself). To implement 'trust', the DRM system is equipped with a 'black box' that performs decryption and encryption functions for such DRM system. The black box includes a public / private key pair, a version number and a unique signature, all as provided by an approved certifying authority. The public key is made available to the license server for purposes of encrypting portions of the issued license, thereby binding such license to such black box. The private key is available to the black box only, and not to the user or anyone else, for purposes of decrypting information encrypted with the corresponding public key. The DRM system is initially provided with a black box with a public / private key pair, and the user is prompted to download from a black box server an updated secure black box when the user first requests a license. The black box server provides the updated black box, along with a unique public/private key pair. Such updated black box is written in unique executable code that will run only on the user's computing device, and is re-updated on a regular basis. When a user requests a license, the client machine sends the black box public key, version number, and signature to the license server, and such license server issues a license only if the version number is current and the signature is valid. A license request also includes an identification of the digital content for which a license is requested and a key ID that identifies the decryption key associated with the requested digital content. The license server uses the black box public key to encrypt the decryption key, and the decryption key to encrypt the license terms, then downloads the encrypted decryption key and encrypted license terms to the user's computing device along with a license signature.

Once the downloaded license has been stored in the DRM system license store, the user can render the digital content according to the rights conferred by the license and specified in the license terms. When a request is made to render the digital content, the black box is caused to decrypt the decryption key and license terms, and a DRM system license evaluator evaluates such license terms. The black box decrypts the encrypted digital content only if the license evaluation results in a decision that the requestor is allowed to play such content. The decrypted content is provided to the rendering application for rendering.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing summary, as well as the following detailed description of the embodiments of the present invention, will be better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there are shown in the drawings embodiments which are presently preferred. As should be understood, however, the invention is not limited to the precise arrangements and instrumentalities shown. In the drawings:

Fig. 1 is a block diagram showing an enforcement architecture in accordance with one embodiment of the present invention;

Fig. 2 is a block diagram of the authoring tool of the architecture of Fig. 1 in accordance with one embodiment of the present invention;

Fig. 3 is a block diagram of a digital content package having digital content for use in connection with the architecture of Fig. 1 in accordance with one embodiment of the present invention;

Fig. 4 is a block diagram of the user's computing device of Fig. 1 in accordance with one embodiment of the present invention;

Figs. 5A and 5B are flow diagrams showing the steps performed in connection with the Digital Rights Management (DRM) system of the computing device of Fig. 4 to render content in accordance with one embodiment of the present invention;

Fig. 6 is a flow diagram showing the steps performed in connection with the DRM system of Fig. 4 to determine whether any valid, enabling licenses are present in accordance with one embodiment of the present invention;

Fig. 7 is a flow diagram showing the steps performed in connection with the DRM system of Fig. 4 to obtain a license in accordance with one embodiment of the present invention;

Fig. 8 is a block diagram of a digital license for use in connection with the architecture of Fig. 1 in accordance with one embodiment of the present invention;

Fig. 9 is a flow diagram showing the steps performed in connection with the DRM system of Fig. 4 to obtain a new black box in accordance with one embodiment of the present invention;

Fig. 10 is a flow diagram showing the key transaction steps performed in connection with the DRM system of Fig. 4 to validate a license and a piece of digital content and render the content in accordance with one embodiment of the present invention;

Fig. 11 is a block diagram showing the license evaluator of Fig. 4 along with a Digital Rights License (DRL) of a license and a language engine for interpreting the DRL in accordance with one embodiment of the present invention; and

Fig. 12 is a block diagram representing a general purpose computer system in which aspects of the present invention and/or portions thereof may be incorporated.

Detailed Description of the Invention

Referring to the drawings in details, wherein like numerals are used to indicate like elements throughout, there is shown in Fig. 1 an enforcement architecture 10 in accordance with one embodiment of the present invention. Overall, the enforcement architecture 10 allows an owner of digital content 12 to specify license rules that must be satisfied before such digital content 12 is allowed to be rendered on a user's computing device 14. Such license rules are embodied within a digital license 16 that the user / user's computing device 14 (hereinafter, such terms are interchangeable unless circumstances require otherwise) must obtain from the content owner or an agent thereof. The digital content 12 is distributed in an encrypted form, and may be distributed freely and widely. Preferably, the decrypting key (KD) for decrypting the digital content 12 is included with the license 16.

COMPUTER ENVIRONMENT

Fig. 12 and the following discussion are intended to provide a brief general description of a suitable computing environment in which the present invention and/or portions thereof may be implemented. Although not required, the invention is described in the general context of computer-executable instructions, such as program modules, being executed by a computer, such as a client workstation or a server. Generally, program modules include routines, programs, objects, components, data structures and the like that perform particular tasks or implement particular abstract data types. Moreover, it should be appreciated that the invention and/or portions thereof may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

As shown in Fig. 12, an exemplary general purpose computing system includes a conventional personal computer 120 or the like, including a processing unit 121, a system memory 122, and a system bus 18 that couples various system components including the system memory to the processing unit 121. The system bus 18 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read-only memory (ROM) 19 and random access memory (RAM) 20. A basic input/output system 21 (BIOS), containing the basic routines that help to transfer information between elements within the personal computer 120, such as during start-up, is stored in ROM 19.

The personal computer 120 may further include a hard disk drive 22 for reading from and writing to a hard disk (not shown), a magnetic disk drive 128 for reading from or writing to a removable magnetic disk 129, and an optical disk drive 25 for reading from or writing to a removable optical disk 131 such as a CD-ROM or other optical media. The hard disk drive 22, magnetic disk drive 128, and optical disk drive 25 are connected to the system bus 18 by a hard disk drive interface 27, a magnetic disk drive interface 28, and an optical drive interface 29, respectively. The drives and their associated computer-readable media provide non-volatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 20.

Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 129, and a removable optical disk 131, it should be appreciated that other types of computer readable media which can store data that is accessible by a computer may also be used in the exemplary operating environment. Such other types of media include a magnetic cassette, a flash memory card, a digital video disk, a Bernoulli cartridge, a random access memory (RAM), a read-only memory (ROM), and the like.

A number of program modules may be stored on the hard disk, magnetic disk 129, optical disk 131, ROM 19 or RAM 20, including an operating system 30, one or more application programs 136, other program modules 137 and program data 138. A user may enter commands and information into the personal computer 120 through input devices such as a keyboard 35 and pointing device 142. Other input devices (not shown) may include a microphone, joystick, game pad, satellite disk, scanner, or the like. These and other input devices are often connected to the processing unit 121 through a serial port interface 41 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port, or universal serial bus (USB). A monitor 42 or other type of display device is also connected to the system bus 18 via an interface, such as a video adapter 148. In addition to the monitor 42, a personal computer typically includes other peripheral output devices (not shown), such as speakers and printers. The exemplary system of Fig. 12 also includes a host adapter 50, a Small Computer System Interface (SCSI) bus 156, and an external storage device 162 connected to the SCSI bus 156.

The personal computer 120 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 149. The remote computer 149 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 120, although only a memory storage device 150 has been illustrated in Fig. 12. The logical connections depicted in Fig. 12 include a local area network (LAN) 46 and a wide area network (WAN) 47. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When used in a LAN networking environment, the personal computer 120 is connected to the LAN 46 through a network interface or adapter 48. When used in a WAN networking environment, the personal computer 120 typically includes a modem 49 or other means for establishing communications over the wide area network 47, such as the Internet. The modem 49, which may be internal or external, is connected to the system bus 18 via the serial port interface 41. In a networked environment, program modules depicted relative to the personal computer 120, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

ARCHITECTURE

Referring again to Fig. 1, in one embodiment of the present invention, the architecture 10 includes an authoring tool 18, a content-key database 20, a content server 22, a license server 24, and a black box server 26, as well as the aforementioned user's computing device 14.

ARCHITECTURE - Authoring Tool 18

The authoring tool 18 is employed by a content owner to package a piece of digital content 12 into a form that is amenable for use in connection with the architecture 10 of the present invention. In particular, the content owner provides the authoring tool 18 with the digital content 12, instructions and/or rules that are to accompany the digital content 12, and instructions and/or rules as to how the digital content 12 is to be packaged. The authoring tool 18 then produces a digital content package 12p having the digital content 12 encrypted according to an encryption / decryption key, and the instructions and/or rules that accompany the digital content 12.

In one embodiment of the present invention, the authoring tool 18 is instructed to serially produce several different digital content 12 packages 12p, each having the same digital content 12 encrypted according to a different encryption / decryption key. As should be understood, having several different packages 12p with the same digital content 12 may be useful for tracking the distribution of such packages 12p / content 12 (hereinafter simply "digital content 12", unless circumstances require otherwise). Such distribution tracking is not ordinarily necessary, but may be used by an investigative authority in cases where the digital content 12 has been illegally sold or broadcast.

In one embodiment of the present invention, the encryption / decryption key that encrypts the digital content 12 is a symmetric key, in that the encryption key is also the decryption key (KD). As will be discussed below in more detail, such decryption key (KD) is delivered to a user's computing device 14 in a hidden form as part of a license 16 for such digital content 12. Preferably, each piece of digital content 12 is provided with a content ID (or each package 12p is provided with a package ID), each decryption key (KD) has a key ID, and the authoring tool 18 causes the decryption key (KD), key ID, and content ID (or package ID) for each piece of digital content 12 (or each package 12p) to be stored in the content-key database 20. In addition, license data regarding the types of licenses 16 to be issued for the digital content 12 and the terms and conditions for each type of license 16 may be stored in the content-key database 20, or else in another database (not shown). Preferably, the license data can be modified by the content owner at a later time as circumstances and market conditions may require.

In use, the authoring tool 18 is supplied with information including, among other things:

- the digital content 12 to be packaged;
- the type and parameters of watermarking and/or fingerprinting to be employed, if any;
- the type and parameters of data compression to be employed, if any;
- the type and parameters of encryption to be employed;
- the type and parameters of serialization to be employed, if any; and
- the instructions and/or rules that are to accompany the digital content 12.

As is known, a watermark is a hidden, computer-readable signal that is added to the digital content 12 as an identifier. A fingerprint is a watermark that is different for each instance. As should be understood, an instance is a version of the digital content 12 that is unique. Multiple copies of any instance may be made, and any copy is of a particular instance. When a specific instance of digital content 12 is illegally sold or broadcast, an investigative authority can perhaps identify suspects according to the watermark / fingerprint added to such digital content 12.

Data compression may be performed according to any appropriate compression algorithm without departing from the spirit and scope of the present invention. For example, the .mp3 or .wav compression algorithm may be employed. Of course, the digital content 12 may already be in a compressed state, in which case no additional compression is necessary.

The instructions and/or rules that are to accompany the digital content 12 may include practically any appropriate instructions, rules, or other information without departing from the spirit and scope of the present invention. As will be discussed below, such accompanying instructions / rules / information are primarily employed by the user and the user's computing device 14 to obtain a license 16 to render the digital content 12. Accordingly, such accompanying instructions / rules / information may include an appropriately formatted license acquisition script or the like, as will be described in more detail below. In addition, or in the alternative, such accompanying instructions / rules / information may include 'preview' information designed to provide a user with a preview of the digital content 12.

With the supplied information, the authoring tool 18 then produces one or more packages 12p corresponding to the digital content 12. Each package 12p may then be stored on the content server 22 for distribution to the world.

In one embodiment of the present invention, and referring now to Fig. 2, the authoring tool 18 is a dynamic authoring tool 18 that receives input parameters which can be specified and operated on. Accordingly, such authoring tool 18 can rapidly produce multiple variations of package 12p for multiple pieces of digital content 12. Preferably, the input parameters are embodied in the form of a dictionary 28, as shown, where the dictionary 28 includes such parameters as:

- the name of the input file 29a having the digital content 12;
- the type of encoding that is to take place
- the encryption / decryption key (KD) to be employed,
- the accompanying instructions / rules / information ('header information') to be packaged with the digital content 12 in the package 12p.
- the type of muxing that is to occur: and

- the name of the output file 29b to which the package 12p based on the digital content 12 is to be written.

As should be understood, such dictionary 28 is easily and quickly modifiable by an operator of the authoring tool 18 (human or machine), and therefore the type of authoring performed by the authoring tool 18 is likewise easily and quickly modifiable in a dynamic manner. In one embodiment of the present invention, the authoring tool 18 includes an operator interface (not shown) displayable on a computer screen to a human operator. Accordingly, such operator may modify the dictionary 28 by way of the interface, and further may be appropriately aided and/or restricted in modifying the dictionary 28 by way of the interface.

In the authoring tool 18, and as seen in Fig. 2, a source filter 18a receives the name of the input file 29a having the digital content 12 from the dictionary 28, and retrieves such digital content 12 from such input file and places the digital content 12 into a memory 29c such as a RAM or the like. An encoding filter 18b then performs encoding on the digital content 12 in the memory 29c to transfer the file from the input format to the output format according to the type of encoding specified in the dictionary 28 (i.e., .wav to .asp, .mp3 to .asp, etc.). and places the encoded digital content 12 in the memory 29c. As shown, the digital content 12 to be packaged (music, e.g.) is received in a compressed format such as the .wav or .mp3 format, and is transformed into a format such as the .asp (active streaming protocol) format. Of course, other input and output formats may be employed without departing from the spirit and scope of the present invention.

Thereafter, an encryption filter 18c encrypts the encoded digital content 12 in the memory 29c according to the encryption / decryption key (KD) specified in the dictionary 28, and places the encrypted digital content 12 in the memory 29c. A header filter 18d then adds the header information specified in the dictionary 28 to the encrypted digital content 12 in the memory 29c.

As should be understood, depending on the situation, the package 12p may include multiple streams of temporally aligned digital content 12 (one stream being shown in Fig. 2), where such multiple streams are multiplexed (i.e., 'muxed'). Accordingly, a mux filter 18e performs muxing on the header information and encrypted digital content 12 in the memory 29c according to the type of muxing specified in the dictionary 28, and places the result in the memory 29c. A file writer filter 18f then retrieves the result from the memory 29c and writes such result to the output file 29b specified in the dictionary 28 as the package 12p.

It should be noted that in certain circumstances, the type of encoding to be performed will not normally change. Since the type of muxing typically is based on the type of encoding, it is likewise the case that the type of muxing will not normally change, either. If this is in fact the case, the dictionary 28 need not include parameters on the type of encoding and/or the type of muxing. Instead, it is only necessary that the type of encoding be 'hardwired' into the encoding filter and/or that the type of muxing be 'hardwired' into the mux filter. Of course, as circumstance require, the authoring tool 18 may not include all of the aforementioned filters, or may include other filters, and any included filter may be hardwired or may perform its function according to parameters specified in the dictionary 28, all without departing from the spirit and scope of the present invention.

Preferably, the authoring tool 18 is implemented on an appropriate computer, processor, or other computing machine by way of appropriate software. The structure and operation of such machine and such software should be apparent based on the disclosure herein and therefore do not require any detailed discussion in the present disclosure.

ARCHITECTURE - Content Server 22

Referring again to Fig. 1, in one embodiment of the present invention, the content server 22 distributes or otherwise makes available for retrieval the packages 12p produced by the authoring tool 18. Such packages 12p may be distributed as requested by the content server 22 by way of any appropriate distribution channel without departing from the spirit and scope of the present invention. For example, such distribution channel may be the Internet or another network, an electronic bulletin board, electronic mail, or the like. In addition, the content server 22 may be employed to copy the packages 12p onto magnetic or optical disks or other storage devices, and such storage devices may then be distributed.

It will be appreciated that the content server 22 distributes packages 12p without regard to any trust or security issues. As discussed below, such issues are dealt with in connection with the license server 24 and the relationship between such license server 24 and the user's computing device 14. In one embodiment of the present invention, the content server 22 freely releases and distributes packages 12p having digital content 12 to any distributee requesting same. However, the content server 22 may also release and distribute such packages 12p in a restricted manner without departing from the spirit and scope of the present invention. For example, the content server 22 may first require payment of a pre-determined distribution fee prior to distribution, or may require that a distributee identify itself, or may indeed make a determination of whether distribution is to occur based on an identification of the distributee.

In addition, the content server 22 may be employed to perform inventory management by controlling the authoring tool 18 to generate a number of different packages 12p in advance to meet an anticipated demand. For example, the server could generate 100 packages 12p based on the same digital content 12, and serve each package 12p 10 times. As supplies of packages 12p dwindle to 20, for example, the content server 22 may then direct the authoring tool 18 to generate 80 additional packages 12p, again for example.

Preferably, the content server 22 in the architecture 10 has a unique public / private key pair (PU-CS, PR-CS) that is employed as part of the process of evaluating a license 16 and obtaining a decryption key (KD) for decrypting corresponding digital content 12, as will be explained in more detail below. As is known, a public / private key pair is an asymmetric key, in that what is encrypted in one of the keys in the key pair can only be decrypted by the other of the keys in the key pair. In a public / private key pair encryption system, the public key may be made known to the world, but the private key should always be held in confidence by the owner of such private key. Accordingly, if the content server 22 encrypts data with its private key (PR-CS), it can send the encrypted data out into the world with its public key (PU-CS) for decryption purposes. Correspondingly, if an external device wants to send data to the content server 22 so that only such content server 22 can decrypt such data, such external device must first obtain the public key of the content server 22 (PU-CS) and then must encrypt the data with such public key. Accordingly, the content server 22 (and only the content server 22) can then employ its private key (PR-CS) to decrypt such encrypted data.

As with the authoring tool 18, the content server 22 is implemented on an appropriate computer, processor, or other computing machine by way of appropriate software. The structure and operation of such machine and such software should be apparent based on the disclosure herein and therefore do not require any detailed discussion in the present disclosure. Moreover, in one embodiment of the present invention, the authoring tool 18 and the content server 22 may reside on a single computer, processor, or other computing machine, each in a separate work space. It should be recognized, moreover, that the content server 22 may in certain circumstances include the authoring tool 18 and/or perform the functions of the authoring tool 18, as discussed above.

Structure of Digital Content Package 12p

Referring now to Fig. 3, in one embodiment of the present invention, the digital content package 12p as distributed by the content server 22 includes:

- the digital content 12 encrypted with the encryption / decryption key (KD), as was discussed above (i.e., (KD(CONTENT)));
- the content ID (or package ID) of such digital content 12 (or package 12p);
- the key ID of the decryption key (KD);
- license acquisition information, preferably in an un-encrypted form; and

- the key KD encrypting the content server 22 public key (PU-CS), signed by the content server 22 private key (PR-CS) (i.e., (KD (PU-CS) S (PR-CS))).

With regard to (KD (PU-CS) S (PR-CS)), it is to be understood that such item is to be used in connection with validating the digital content 12 and/or package 12p, as will be explained below. Unlike a certificate with a digital signature (see below), the key (PU-CS) is not necessary to get at (KD (PU-CS)). Instead, the key (PU-CS) is obtained merely by applying the decryption key (KD). Once so obtained, such key (PU-CS) may be employed to test the validity of the signature (S (PR-CS)).

It should also be understood that for such package 12p to be constructed by the authoring tool 18, such authoring tool 18 must already possess the license acquisition information and (KD (PU-CS) S (PR-CS)), presumably as header information supplied by the dictionary 28. Moreover, the authoring tool 18 and the content server 22 must presumably interact to construct (KD (PU-CS) S (PR-CS)). Such interaction may for example include the steps of:

- the content server 22 sending (PU-CS) to the authoring tool 18;
- the authoring tool 18 encrypting (PU-CS) with (KD) to produce (KD (PU-CS));
- the authoring tool 18 sending (KD (PU-CS)) to the content server 22;
- the content server 22 signing (KD (PU-CS)) with (PR-CS) to produce (KD (PU-CS) S (PR-CS)); and
- the content server 22 sending (KD (PU-CS) S (PR-CS)) to the authoring tool 18.

ARCHITECTURE - License Server 24

Referring again to Fig. 1, in one embodiment of the present invention, the license server 24 performs the functions of receiving a request for a license 16 from a user's computing device 14 in connection with a piece of digital content 12, determining whether the user's computing device 14 can be trusted to honor an issued license 16, negotiating such a license 16, constructing such license 16, and transmitting such license 16 to the user's computing device 14. Preferably, such transmitted license 16 includes the decryption key (KD) for decrypting the digital content 12. Such license server 24 and such functions will be explained in more detail below. Preferably, and like the content server 22, the license server 24 in the architecture 10 has a unique public / private key pair (PU-LS, PR-LS) that is employed as part of the process of evaluating a license 16 and obtaining a decryption key (KD) for decrypting corresponding digital content 12, as will be explained in more detail below.

As with the authoring tool 18 and the content server 22, the license server 24 is implemented on an appropriate computer, processor, or other computing machine by way of appropriate software. The structure and operation of such machine and such software should be apparent based on the disclosure herein and therefore do not require any detailed discussion in the present disclosure. Moreover, in one embodiment of the present invention the authoring tool 18 and/or the content server 22 may reside on a single computer, processor, or other computing machine together with the license server 24, each in a separate work space.

In one embodiment of the present invention, prior to issuance of a license 16, the license server 24 and the content server 22 enter into an agency agreement or the like, wherein the license server 24 in effect agrees to be the licensing authority for at least a portion of the digital content 12 distributed by the content server 22. As should be understood, one content server 22 may enter into an agency agreement or the like with several license servers 24, and/or one license server 24 may enter into an agency agreement or the like with several content servers 22, all without departing from the spirit and scope of the present invention.

Preferably, the license server 24 can show to the world that it does in fact have the authority to issue a license 16 for digital content 12 distributed by the content server 22. To do so, it is preferable that the license server 24 send to the content server 22 the license server 24 public key (PU-LS), and that the content server

22 then send to the license server 24 a digital certificate containing PU-LS as the contents signed by the content server 22 private key (CERT (PU-LS) S (PR-CS)). As should be understood, the contents (PU-LS) in such certificate can only be accessed with the content server 22 public key (PU-CS). As should also be understood, in general, a digital signature of underlying data is an encrypted form of such data, and will not match such data when decrypted if such data has been adulterated or otherwise modified.

As a licensing authority in connection with a piece of digital content 12, and as part of the licensing function, the license server 24 must have access to the decryption key (KD) for such digital content 12. Accordingly, it is preferable that license server 24 have access to the content-key database 20 that has the decryption key (KD), key ID, and content ID (or package ID) for such digital content 12 (or package 12p).

ARCHITECTURE - Black Box Server 26

Still referring to Fig. 1, in one embodiment of the present invention, the black box server 26 performs the functions of installing and/or upgrading a new black box 30 in a user's computing device 14. As will be explained in more detail below, the black box 30 performs encryption and decryption functions for the user's computing device 14. As will also be explained in more detail below, the black box 30 is intended to be secure and protected from attack. Such security and protection is provided, at least in part, by upgrading the black box 30 to a new version as necessary by way of the black box server 26, as will be explained in more detail below.

As with the authoring tool 18, the content server 22, and the license server 24, the black box server 26 is implemented on an appropriate computer, processor, or other computing machine by way of appropriate software. The structure and operation of such machine and such software should be apparent based on the disclosure herein and therefore do not require any detailed discussion in the present disclosure. Moreover, in one embodiment of the present invention the license server 24, the authoring tool 18, and/or the content server 22 may reside on a single computer, processor, or other computing machine together with the black box server 26, each in a separate work space. Note, though, that for security purposes. it may be wise to have the black box server 26 on a separate machine.

ARCHITECTURE - User's Computing Device 14

Referring now to Fig. 4, in one embodiment of the present invention, the user's computing device 14 is a personal computer or the like, having elements including a keyboard, a mouse, a screen, a processor, RAM. ROM, a hard drive, a floppy drive, a CD player, and/or the like. However, the user's computing device 14 may also be a dedicated viewing device such as a television or monitor, a dedicated audio device such as a stereo or other music player, a dedicated printer, or the like, among other things, all without departing from the spirit and scope of the present invention.

The content owner for a piece of digital content 12 must trust that the user's computing device 14 will abide by the rules specified by such content owner, i.e. that the digital content 12 will not be rendered unless the user obtains a license 16 that permits the rendering in the manner sought. Preferably, then, the user's computing device 14 must provide a trusted component or mechanism 32 that can satisfy to the content owner that such computing device 14 will not render the digital content 12 except according to the license rules embodied in the license 16 associated with the digital content 12 and obtained by the user.

Here, the trusted mechanism 32 is a Digital Rights Management (DRM) system 32 that is enabled when a user requests that a piece of digital content 12 be rendered, that determines whether the user has a license 16 to render the digital content 12 in the manner sought, that effectuates obtaining such a license 16 if necessary, that determines whether the user has the right to play the digital content 12 according to the license 16, and that decrypts the digital content 12 for rendering purposes if in fact the user has such right according to such license 16. The contents and function of the DRM system 32 on the user's computing device 14 and in connection with the architecture 10 are described below.

DRM SYSTEM 32

The DRM system 32 performs four main functions with the architecture 10 disclosed herein: (1) content acquisition, (2) license acquisition, (3) content rendering, and (4) black box 30 installation / update. Preferably, any of the functions can be performed at any time, although it is recognized that some of the functions already require that digital content 12 be acquired.

DRM SYSTEM 32 - Content Acquisition

Acquisition of digital content 12 by a user and/or the user's computing device 14 is typically a relatively straight-forward matter and generally involves placing a file having encrypted digital content 12 on the user's computing device 14. Of course, to work with the architecture 10 and the DRM system 32 disclosed herein, it is necessary that the encrypted digital content 12 be in a form that is amenable to such architecture 10 and DRM system 32, such as the digital package 12p as will be described below.

As should be understood, the digital content 12 may be obtained in any manner from a content server 22, either directly or indirectly, without departing from the spirit and scope of the present invention. For example, such digital content 12 may be downloaded from a network such as the Internet, located on an obtained optical or magnetic disk or the like, received as part of an E-mail message or the like, or downloaded from an electronic bulletin board or the like.

Such digital content 12, once obtained, is preferably stored in a manner such that the obtained digital content 12 is accessible by a rendering application 34 (to be described below) running on the computing device 14, and by the DRM system 32. For example, the digital content 12 may be placed as a file on a hard drive (not shown) of the user's computing device 14, or on a network server (not shown) accessible to the computing device 14. In the case where the digital content 12 is obtained on an optical or magnetic disk or the like, it may only be necessary that such disk be present in an appropriate drive (not shown) coupled to the user's computing device 14.

In the present invention, it is not envisioned that any special tools are necessary to acquire digital content 12. either from the content server 22 as a direct distribution source or from some intermediary as an indirect distribution source. That is, it is preferable that digital content 12 be as easily acquired as any other data file. However, the DRM system 32 and/or the rendering application 34 may include an interface (not shown) designed to assist the user in obtaining digital content 12 . For example, the interface may include a web browser especially designed to search for digital content 12, links to pre-defined Internet web sites that are known to be sources of digital content 12, and the like.

DRM SYSTEM 32 - Content Rendering, Part 1

Referring now to Fig. 5A, in one embodiment of the present invention, assuming the encrypted digital content 12 has been distributed to and received by a user and placed by the user on the computing device 14 in the form of a stored file, the user will attempt to render the digital content 12 by executing some variation on a render command (step 501). For example, such render command may be embodied as a request to 'play' or 'open' the digital content 12. In some computing environments, such as for example the "MICROSOFT WINDOWS" operating system, distributed by MICROSOFT Corporation of Redmond, Washington. such play or open command may be as simple as 'clicking' on an icon representative of the digital content 12. Of course, other embodiments of such render command may be employed without departing from the spirit and scope of the present invention. In general, such render command may be considered to be executed whenever a user directs that a file having digital content 12 be opened, run, executed, and/or the like.

Importantly, and in addition. such render command may be embodied as a request to copy the digital content 12 to another form. such as to a printed form, a visual form, an audio form, etc. As should be understood. the same digital content 12 may be rendered in one form, such as on a computer screen, and then in another form, such as a printed document. In the present invention. each type of rendering is performed only if the user has the right to do so. as will be explained below.

In one embodiment of the present invention. the digital content 12 is in the form of a digital file having a file name ending with an extension, and the computing device 14 can determine based on such extension to start a particular kind of rendering application 34. For example, if the file name extension indicates that the digital content 12 is a text file, the rendering application 34 is some form of word processor such as the "MICROSOFT WORD", distributed by MICROSOFT Corporation of Redmond, Washington. Likewise, if the file name extension indicates that the digital content 12 is an audio, video, and/or multimedia file, the rendering application 34 is some form of multimedia player, such as "MICROSOFT MEDIA PLAYER", also distributed by MICROSOFT Corporation of Redmond, Washington.

Of course, other methods of determining a rendering application may be employed without departing from the spirit and scope of the present invention. As but one example, the digital content 12 may contain meta-data in an un-encrypted form (i.e., the aforementioned header information), where the meta-data includes information on the type of rendering application 34 necessary to render such digital content 12.

Preferably, such rendering application 34 examines the digital content 12 associated with the file name and determines whether such digital content 12 is encrypted in a rights-protected form (steps 503, 505). If not protected, the digital content 12 may be rendered without further ado (step 507). If protected, the rendering application 34 determines from the encrypted digital content 12 that the DRM system 32 is necessary to play such digital content 12. Accordingly, such rendering application 34 directs the user's computing device 14 to run the DRM system 32 thereon (step 509). Such rendering application 34 then calls such DRM system 32 to decrypt the digital content 12 (step 511). As will be discussed in more detail below, the DRM system 32 in fact decrypts the digital content 12 only if the user has a valid license 16 for such digital content 12 and the right to play the digital content 12 according to the license rules in the valid license 16. Preferably, once the DRM system 32 has been called by the rendering application 34, such DRM system 32 assumes control from the rendering application 34, at least for purposes of determining whether the user has a right to play such digital content 12 (step 513).

DRM System 32 Components

In one embodiment of the present invention, and referring again to Fig. 4, the DRM system 32 includes a license evaluator 36, the black box 30, a license store 38, and a state store 40.

DRM System 32 Components - License Evaluator 36

The license evaluator 36 locates one or more licenses 16 that correspond to the requested digital content 12, determines whether such licenses 16 are valid, reviews the license rules in such valid licenses 16, and determines based on the reviewed license rules whether the requesting user has the right to render the requested digital content 12 in the manner sought, among other things. As should be understood, the license evaluator 36 is a trusted component in the DRM system 32. In the present disclosure, to be 'trusted' means that the license server 24 (or any other trusting element) is satisfied that the trusted element will carry out the wishes of the owner of the digital content 12 according to the rights description in the license 16, and that a user cannot easily alter such trusted element for any purpose, nefarious or otherwise.

The license evaluator 36 has to be trusted in order to ensure that such license evaluator 36 will in fact evaluate a license 16 properly, and to ensure that such license evaluator 36 has not been adulterated or otherwise modified by a user for the purpose of bypassing actual evaluation of a license 16. Accordingly, the license evaluator 36 is run in a protected or shrouded environment such that the user is denied access to such license evaluator 36. Other protective measures may of course be employed in connection with the license evaluator 36 without departing from the spirit and scope of the present invention.

DRM System 32 Components - Black Box 30

Primarily, and as was discussed above, the black box 30 performs encryption and decryption functions in the DRM system 32. In particular, the black box 30 works in conjunction with the license evaluator 36 to decrypt and encrypt certain information as part of the license evaluation function. In addition, once the license evaluator 36 determines that a user does in fact have the right to render the requested digital content 12 in the manner sought, the black box 30 is provided with a decryption key (KD) for such digital content 12, and performs the function of decrypting such digital content 12 based on such decryption key (KD).

The black box 30 is also a trusted component in the DRM system 32. In particular, the license server 24 must trust that the black box 30 will perform the decryption function only in accordance with the license rules in the license 16, and also trust that such black box 30 will not operate should it become adulterated or otherwise modified by a user for the nefarious purpose of bypassing actual evaluation of a license 16. Accordingly, the black box 30 is also run in a protected or shrouded environment such that the user is denied access to such black box 30. Again, other protective measures may be employed in connection with the black box 30 without departing from the spirit and scope of the present invention. Preferably, and like the content server 22 and license server 24, the black box 30 in the DRM system 32 has a unique public / private key pair (PU-BB, PR-BB) that is employed as part of the process of evaluating the license 16 and obtaining a decryption key (KD) for decrypting the digital content 12, as will be described in more detail below.

DRM System 32 Components - License Store 38

The license store 38 stores licenses 16 received by the DRM system 32 for corresponding digital content 12. The license store 38 itself need not be trusted since the license store 38 merely stores licenses 16, each of which already has trust components built thereinto, as will be described below. In one embodiment of the present invention, the license store 38 is merely a sub-directory of a drive such as a hard disk drive or a network drive. However, the license store 38 may be embodied in any other form without departing from the spirit and scope of the present invention, so long as such license store 38 performs the function of storing licenses 16 in a location relatively convenient to the DRM system 32.

DRM System 32 Components - State Store 40

The state store 40 performs the function of maintaining state information corresponding to licenses 16 presently or formerly in the license store 38. Such state information is created by the DRM system 32 and stored in the state store 40 as necessary. For example, if a particular license 16 only allows a pre-determined number of renderings of a piece of corresponding digital content 12, the state store 40 maintains state information on how many renderings have in fact taken place in connection with such license 16. The state store 40 continues to maintain state information on licenses 16 that are no longer in the license store 38 to avoid the situation where it would otherwise be advantageous to delete a license 16 from the license store 38 and then obtain an identical license 16 in an attempt to delete the corresponding state information from the state store 40.

The state store 40 also has to be trusted in order to ensure that the information stored therein is not reset to a state more favorable to a user. Accordingly, the state store 40 is likewise run in a protected or shrouded environment such that the user is denied access to such state store 40. Once again, other protective measures may of course be employed in connection with the state store 40 without departing from the spirit and scope of the present invention. For example, the state store 40 may be stored by the DRM system 32 on the computing device 14 in an encrypted form.

DRM SYSTEM 32 - Content Rendering, Part 2

Referring again to Fig. 5A, and again discussing content rendering in one embodiment of the present invention, once the DRM system 32 has assumed control from the calling rendering application 34, such DRM system 32 then begins the process of determining whether the user has a right to render the requested digital content 12 in the manner sought. In particular, the DRM system 32 either locates a valid, enabling license 16 in the license store (steps 515, 517) or attempts to acquire a valid, enabling license 16 from the license server 24 (i.e. performs the license acquisition function as discussed below and as shown in Fig. 7).

As a first step, and referring now to Fig. 6, the license evaluator 36 of such DRM system 32 checks the license store 38 for the presence of one or more received licenses 16 that correspond to the digital content 12 (step 601). Typically, the license 16 is in the form of a digital file, as will be discussed below, although it will be recognized that the license 16 may also be in other forms without departing from the spirit and scope of the present invention. Typically, the user will receive the digital content 12 without such license 16, although it will likewise be recognized that the digital content 12 may be received with a corresponding license 16 without departing from the spirit and scope of the present invention.

As was discussed above in connection with Fig. 3, each piece of digital content 12 is in a package 12p with a content ID (or package ID) identifying such digital content 12 (or package 12p), and a key ID identifying the decryption key (KD) that will decrypt the encrypted digital content 12. Preferably, the content ID (or package ID) and the key ID are in an un-encrypted form. Accordingly, and in particular, based on the content ID of the digital content 12, the license evaluator 36 looks for any license 16 in the license store 38 that contains an identification of applicability to such content ID. Note that multiple such licenses 16 may be found, especially if the owner of the digital content 12 has specified several different kinds of licenses 16 for such digital content 12, and the user has obtained multiple ones of such licenses 16. If in fact the license evaluator 36 does not find in the license store 38 any license 16 corresponding to the requested digital content 12, the DRM system 32 may then perform the function of license acquisition (step 519 of Fig. 5), to be described below.

Assume now that the DRM system 32 has been requested to render a piece of digital content 12, and one or more licenses 16 corresponding thereto are present in the license store 38. In one embodiment of the present invention, then, the license evaluator 36 of the DRM system 32 proceeds to determine for each such license 16 whether such license 16 itself is valid (steps 603 and 605 of Fig. 6). Preferably, and in particular, each license 16 includes a digital signature 26 based on the content 28 of the license 16. As should be understood, the digital signature 26 will not match the license 16 if the content 28 has been adulterated or otherwise modified. Thus, the license evaluator 36 can determine based on the digital signature 26 whether the content 28 is in the form that it was received from the license server 24 (i.e., is valid). If no valid license 16 is found in the license store 38, the DRM system 32 may then perform the license acquisition function described below to obtain such a valid license 16.

Assuming that one or more valid licenses 16 are found, for each valid license 16, the license evaluator 36 of the DRM system 32 next determines whether such valid license 16 gives the user the right to render the corresponding digital content 12 in the manner desired (i.e., is enabling) (steps 607 and 609). In particular, the license evaluator 36 determines whether the requesting user has the right to play the requested digital content 12 based on the rights description in each license 16 and based on what the user is attempting to do with the digital content 12. For example, such rights description may allow the user to render the digital content 12 into a sound, but not into a decrypted digital copy.

As should be understood, the rights description in each license 16 specifies whether the user has rights to play the digital content 12 based on any of several factors, including who the user is, where the user is located, what type of computing device 14 the user is using, what rendering application 34 is calling the DRM system 32, the date, the time, etc. In addition, the rights description may limit the license 16 to a pre-determined number of plays, or pre-determined play time, for example. In such case, the DRM system 32 must refer to any state information with regard to the license 16, (i.e., how many times the digital content 12 has been rendered, the total amount of time the digital content 12 has been rendered, etc.), where such state information is stored in the state store 40 of the DRM system 32 on the user's computing device 14.

Accordingly, the license evaluator 36 of the DRM system 32 reviews the rights description of each valid license 16 to determine whether such valid license 16 confers the rights sought to the user. In doing so, the license evaluator 36 may have to refer to other data local to the user's computing device 14 to perform a determination of whether the user has the rights sought. As seen in Fig. 4, such data may include an identification 42 of the user's computing device (machine) 14 and particular aspects thereof, an identification 44 of the user and particular aspects thereof, an identification of the rendering application 34 and particular aspects thereof, a system clock 46, and the like. If no valid license 16 is found that provides the user with the right to render the digital content 12 in the manner sought. the DRM system 32 may then perform the license acquisition function described below to obtain such a license 16, if in fact such a license 16 is obtainable.

Of course, in some instances the user cannot obtain the right to render the digital content 12 in the manner requested, because the content owner of such digital content 12 has in effect directed that such right not be granted. For example, the content owner of such digital content 12 may have directed that no license 16 be granted to allow a user to print a text document, or to copy a multimedia presentation into an un-encrypted form. In one embodiment of the present invention, the digital content 12 includes data on what rights are available upon purchase of a license 16, and types of licenses 16 available. However, it will be recognized that the content owner of a piece of digital content 12 may at any time change the rights currently available for such digital content 12 by changing the licenses 16 available for such digital content 12.

DRM SYSTEM 32 - License Acquisition

Referring now to Fig. 7, if in fact the license evaluator 36 does not find in the license store 38 any valid, enabling license 16 corresponding to the requested digital content 12, the DRM system 32 may then perform the function of license acquisition. As shown in Fig. 3, each piece of digital content 12 is packaged with information in an un-encrypted form regarding how to obtain a license 16 for rendering such digital content 12 (i.e., license acquisition information).

In one embodiment of the present invention. such license acquisition information may include (among other things) types of licenses 16 available, and one or more Internet web sites or other site information at which one or more appropriate license servers 24 may be accessed, where each such license server 24 is in fact capable of issuing a license 16 corresponding to the digital content 12. Of course, the license 16 may be obtained in other manners without departing from the spirit and scope of the present invention. For example, the license 16 may be obtained from a license server 24 at an electronic bulletin board, or even in person or via regular mail in the form of a file on a magnetic or optical disk or the like.

Assuming that the location for obtaining a license 16 is in fact a license server 24 on a network, the license evaluator 36 then establishes a network connection to such license server 24 based on the web site or other site information, and then sends a request for a license 16 from such connected license server 24 (steps 701, 703). In particular, once the DRM system 32 has contacted the license server 24, such DRM system 32 transmits appropriate license request information 36 to such license server 24. In one embodiment of the present invention, such license 16 request information 36 may include:

- the public key of the black box 30 of the DRM system 32 (PU-BB);
- the version number of the black box 30 of the DRM system 32;
- a certificate with a digital signature from a certifying authority certifying the black box 30 (where the certificate may in fact include the aforementioned public key and version number of the black box 30);
- the content ID (or package ID) that identifies the digital content 12 (or package 12p);
- the key ID that identifies the decryption key (KD) for decrypting the digital content 12;
- the type of license 16 requested (if in fact multiple types are available);
- the type of rendering application 34 that requested rendering of the digital content 12;

and/or the like, among other things. Of course, greater or lessor amounts of license 16 request information 36 may be transmitted to the license server 24 by the DRM system

32 without departing from the spirit and scope of the present invention. For example, information on the type of rendering application 34 may not be necessary, while additional information about the user and/or the user's computing device 14 may be necessary.

Once the license server 24 has received the license 16 request information 36 from the DRM system 32, the license server 24 may then perform several checks for trust / authentication and for other purposes. In one embodiment of the present invention, such license server 24 checks the certificate with the digital signature of the certifying authority to determine whether such has been adulterated or otherwise modified (steps 705, 707). If so, the license server 24 refuses to grant any license 16 based on the request information 36. The license server 24 may also maintain a list of known 'bad' users and/or user's computing devices 14, and may refuse to grant any license 16 based on a request from any such bad user and/or bad user's computing device 14 on the list. Such 'bad' list may be compiled in any appropriate manner without departing from the spirit and scope of the present invention.

Based on the received request and the information associated therewith, and particularly based on the content ID (or package ID) in the license request information, the license server 24 can interrogate the content-key database 20 (Fig. 1) and locate a record corresponding to the digital content 12 (or package 12p) that is the basis of the request. As was discussed above, such record contains the decryption key (KD), key ID, and content ID for such digital content 12. In addition, such record may contain license data regarding the types of licenses 16 to be issued for the digital content 12 and the terms and conditions for each type of license 16. Alternatively, such record may include a pointer, link, or reference to a location having such additional information.

As mentioned above, multiple types of licenses 16 may be available. For example, for a relatively small license fee, a license 16 allowing a limited number of renderings may be available. For a relatively greater license fee, a license 16 allowing unlimited renderings until an expiration date may be available. For a still greater license fee, a license 16 allowing unlimited renderings without any expiration date may be available. Practically any type of license 16 having any kind of license terms may be devised and issued by the license server 24 without departing from the spirit and scope of the present invention.

In one embodiment of the present invention, the request for a license 16 is accomplished with the aid of a web page or the like as transmitted from the license server 24 to the user's computing device 14. Preferably, such web page includes information on all types of licenses 16 available from the license server 24 for the digital content 12 that is the basis of the license 16 request.

In one embodiment of the present invention, prior to issuing a license 16, the license server 24 checks the version number of the black box 30 to determine whether such black box 30 is relatively current (steps 709, 711). As should be understood, the black box 30 is intended to be secure and protected from attacks from a user with nefarious purposes (i.e., to improperly render digital content 12 without a license 16, or outside the terms of a corresponding license 16). However, it is to be recognized that no system and no software device is in fact totally secure from such an attack.

As should also be understood, if the black box 30 is relatively current, i.e., has been obtained or updated relatively recently, it is less likely that such black box 30 has been successfully attacked by such a nefarious user. Preferably, and as a matter of trust, if the license server 24 receives a license request with request information 36 including a black box 30 version number that is not relatively current, such license server 24 refuses to issue the requested license 16 until the corresponding black box 30 is upgraded to a current version, as will be described below. Put simply, the license server 24 will not trust such black box 30 unless such black box 30 is relatively current.

In the context of the black box 30 of the present invention, the term 'current' or 'relatively current' may have any appropriate meaning without departing from the spirit and scope of the present invention. consistent with the function of providing trust in the black box 30 based on the age or use thereof. For example, 'current' may be defined according to age (i.e., less than one month old). As an alternative example, 'current' may be defined based on a number of times that the black box 30 has decrypted digital content 12 (i.e., less than 200 instances of decryption). Moreover, 'current' may be based on policy as set by each license server 24, where one license server 24 may define 'current' differently from another license server 24, and a license server 24 may further define 'current' differently depending on the digital content 12 for which a license 16 is requested. or depending on the type of license 16 requested, among other things.

Assuming that the license server 24 is satisfied from the version number of a black box 30 or other indicia thereof that such black box 30 is current, the license server 24 then proceeds to negotiate terms and conditions for the license 16 with the user (step 713). Alternatively, the license server 24 negotiates the license 16 with the user, then satisfies itself from the version number of the black box 30 that such black box 30 is current (i.e., performs step 713, then step 711). Of course, the amount of negotiation varies depending on the type of license 16 to be issued, and other factors. For example, if the license server 24 is merely issuing a paid-up unlimited use license 16, very little need be negotiated. On the other hand, if the license 16 is to be based on such items as varying values, sliding scales, break points, and other details, such items and details may need to be worked out between the license server 24 and the user before the license 16 can be issued.

As should be understood, depending on the circumstances, the license negotiation may require that the user provide further information to the license server 24 (for example, information on the user, the user's computing device 14, etc.). Importantly, the license negotiation may also require that the user and the license server 24 determine a mutually acceptable payment instrument (a credit account, a debit account, a mailed check, etc.) and/or payment method (paid-up immediately, spread over a period of time, etc.), among other things.

Once all the terms of the license 16 have been negotiated and agreed to by both the license server 24 and user (step 715). a digital license 16 is generated by the license server 24 (step 719), where such generated license 16 is based at least in part on the license request, the black box 30 public key (PU-BB). and the decryption key (KD) for the digital content 12 that is the basis of the request as obtained from the content-key database 20. In one embodiment of the present invention, and as seen in Fig. 8, the generated license 16 includes:

- the content ID of the digital content 12 to which the license 16 applies;
- a Digital Rights License (DRL) 48 (i.e., the rights description or actual terms and conditions of the license 16 written in a predetermined form that the license evaluator 36 can interrogate), perhaps encrypted with the decryption key (KD) (i.e.. KD (DRL));
- the decryption key (KD) for the digital content 12 encrypted with the black box 30 public key (PU-BB) as receive in the license request (i.e.,(PU-BB (KD));
- a digital signature from the license server 24 (without any attached certificate) based on (KD (DRL)) and (PU-BB (KD)) and encrypted with the license server 24 private key (i.e.. (S (PR-LS))); and
- the certificate that the license server 24 obtained previously from the content server 22, such certificate indicating that the license server 24 has the authority from the content server 22 to issue the license 16 (i.e., (CERT (PU-LS) S (PR-CS))).

As should be understood, the aforementioned elements and perhaps others are packaged into a digital file or some other appropriate form. As should also be understood, if the DRL 48 or (PU-BB (KD)) in the license 16 should become adulterated or otherwise modified, the digital signature (S (PR-LS)) in the license 16 will not match and therefore will not validate such license 16. For this reason. the DRL 48 need not necessarily be in an encrypted form (i.e.. (KD(DRL)) as mentioned above), although such encrypted form may in some instances be desirable and therefore may be employed without departing from the spirit and scope of the present invention.

Once the digital license 16 has been prepared. such license 16 is then issued to the requestor (i.e., the DRM system 32 on the user's computing device 14) (step 719 of Fig. 7). Preferably, the license 16 is transmitted over the same path through which the request therefor was made (i.e., the Internet or another network), although another path may be employed without departing from the spirit and scope of the present invention. Upon receipt, the requesting DRM system 32 preferably automatically places the received digital license 16 in the license store 38 (step 721).

It is to be understood that a user's computing device 14 may on occasion malfunction, and licenses 16 stored in the license store 38 of the DRM system 32 on such user's computing device 14 may become irretrievably lost. Accordingly, it is preferable that the license server 24 maintain a database 50 of issued licenses 16 (Fig. 1), and that such license server 24 provide a user with a copy or re-issue (hereinafter 're-issue') of an issued license 16 if the user is in fact entitled to such re-issue. In the aforementioned case where licenses 16 are irretrievably lost, it is also likely the case that state information stored in the state store 40 and corresponding to such licenses 16 is also lost. Such lost state information should be taken into account when re-issuing a license 16. For example. a fixed number of renderings license 16 might legitimately be re-issued in a pro-rated form after a relatively short period of time, and not re-issued at all after a relatively longer period of time.

DRM SYSTEM 32 - Installation/Upgrade of Black Box 30

As was discussed above, as part of the function of acquiring a license 16, the license server 24 may deny a request for a license 16 from a user if the user's computing device 14 has a DRM system 32 with a black box 30 that is not relatively current, i.e., has a relatively old version number. In such case. it is preferable that the black box 30 of such DRM system 32 be upgraded so that the license acquisition function can then proceed. Of course, the black box 30 may be upgraded at other times without departing from the spirit and scope of the present invention.

Preferably, as part of the process of installing the DRM system 32 on a user's computing device 14, a non-unique 'lite' version of a black box 30 is provided. Such 'lite' black box 30 is then upgraded to a unique regular version prior to rendering a piece of digital content 12. As should be understood, if each black box 30 in each DRM system 32 is unique, a security breach into one black box 30 cannot easily be replicated with any other black box 30.

Referring now to Fig. 9, the DRM system 32 obtains the unique black box 30 by requesting same from a black box server 26 or the like (as was discussed above and as shown in Fig. 1) (step 901). Typically, such request is made by way of the Internet, although other means of access may be employed without departing from the spirit and scope of the present invention. For example, the connection to a black box server 26 may be a direct connection, either locally or remotely. An upgrade from one unique non-lite black box 30 to another unique non-lite black box 30 may also be requested by the DRM system 32 at any time, such as for example a time when a license server 24 deems the black box 30 not current, as was discussed above.

Thereafter, the black box server 26 generates a new unique black box 30 (step 903). As seen in Fig. 3, each new black box 30 is provided with a version number and a certificate with a digital signature from a certifying authority. As was discussed above in connection with the license acquisition function, the version number of the black box 30 indicates the relative age and/or use thereof. The certificate with the digital signature from the certifying authority, also discussed above in connection with the license acquisition function, is a proffer or vouching mechanism from the certifying authority that a license server 24 should trust the black box 30. Of course, the license server 24 must trust the certifying authority to issue such a certificate for a black box 30 that is in fact trustworthy. It may be the case, in fact, that the license server 24 does not trust a particular certifying authority, and refuses to honor any certificate issued by such certifying authority. Trust may not occur, for example, if a particular certifying authority is found to be engaging in a pattern of improperly issuing certificates.

Preferably, and as was discussed above, the black box server 26 includes a new unique public / private key pair (PU-BB, PR-BB) with the newly generated unique black box 30 (step 903 of Fig. 9). Preferably, the private key for the black box 30 (PR-BB) is accessible only to such black box 30, and is hidden from and inaccessible by the remainder of the world, including the computing device 14 having the DRM system 32 with such black box 30, and the user thereof.

Most any hiding scheme may be employed without departing from the spirit and scope of the present invention, so long as such hiding scheme in fact performs the function of hiding the private key (PR-BB) from the world. As but one example, the private key (PR-BB) may be split into several sub-components, and each sub-component may be encrypted uniquely and stored in a different location. In such a situation, it is preferable that such sub-components are never assembled in full to produce the entire private key (PR-BB).

In one embodiment of the present invention, such private key (PR-BB) is encrypted according to code-based encryption techniques. In particular, in such embodiment, the actual software code of the black box 30 (or other software code) is employed as encrypting key(s). Accordingly, if the code of the black box 30 (or the other software code) becomes adulterated or otherwise modified, for example by a user with nefarious purposes, such private key (PR-BB) cannot be decrypted.

Although each new black box 30 is delivered with a new public / private key pair (PU-BB, PR-BB), such new black box 30 is also preferably given access to old public / private key pairs from old black boxes 30 previously delivered to the DRM system 32 on the user's computing device 14 (step 905). Accordingly, the upgraded black box 30 can still employ the old key pairs to access older digital content 12 and older corresponding licenses 16 that were generated according to such old key pairs, as will be discussed in more detail below.

Preferably, the upgraded black box 30 delivered by the black box server 26 is tightly tied to or associated with the user's computing device 14. Accordingly, the upgraded black box 30 cannot be operably transferred among multiple computing devices 14 for nefarious purposes or otherwise. In one embodiment of the present invention, as part of the request for the black box 30 (step 901) the DRM system 32 provides hardware information unique to such DRM system 32 and/or unique to the user's computing device 14 to the black box server 26, and the black box server 26 generates a black box 30 for the DRM system 32 based in part on such provided hardware information. Such generated upgraded black box 30 is then delivered to and installed in the DRM system 32 on the user's computing device 14 (steps 907, 909). If the upgraded black box 30 is then somehow transferred to another computing device 14, the transferred black box 30 recognizes that it is not intended for such other computing device 14, and does not allow any requested rendering to proceed on such other computing device 14.

Once the new black box 30 is installed in the DRM system 32, such DRM system 32 can proceed with a license acquisition function or with any other function.

DRM SYSTEM 32 - Content Rendering, Part 3

Referring now to Fig. 5B, and assuming, now, that the license evaluator 36 has found at least one valid license 16 and that at least one of such valid licenses 16 provides the user with the rights necessary to render the corresponding digital content 12 in the manner sought (i.e., is enabling), the license evaluator 36 then selects one of such licenses 16 for further use (step 519). Specifically, to render the requested digital content 12, the license evaluator 36 and the black box 30 in combination obtain the decryption key (KD) from such license 16, and the black box 30 employs such decryption key (KD) to decrypt the digital content 12. In one embodiment of the present invention, and as was discussed above, the decryption key (KD) as obtained from the license 16 is encrypted with the black box 30 public key (PU-BB(KD)), and the black box 30 decrypts such encrypted decryption key with its private key (PR-BB) to produce the decryption key (KD) (steps 521, 523). However, other methods of obtaining the decryption key (KD) for the digital content 12 may be employed without departing from the spirit and scope of the present invention.

Once the black box 30 has the decryption key (KD) for the digital content 12 and permission from the license evaluator 36 to render the digital content 12, control may be returned to the rendering application 34 (steps 525, 527). In one embodiment of the present invention, the rendering application 34 then calls the DRM system 32 / black box 30 and directs at least a portion of the encrypted digital content 12 to the black box 30 for decryption according to the decryption key (KD) (step 529). The black box 30 decrypts the digital content 12 based upon the decryption key (KD) for the digital content 12, and then the black box 30 returns the decrypted digital content 12 to the rendering application 34 for actual rendering (steps 533, 535). The rendering application 34 may either send a portion of the encrypted digital content 12 or the entire digital content 12 to the black box 30 for decryption based on the decryption key (KD) for such digital content 12 without departing from the spirit and scope of the present invention.

Preferably, when the rendering application 34 sends digital content 12 to the black box 30 for decryption, the black box 30 and/or the DRM system 32 authenticates such rendering application 34 to ensure that it is in fact the same rendering application 34 that initially requested the DRM system 32 to run (step 531). Otherwise, the potential exists that rendering approval may be obtained improperly by basing the rendering request on one type of rendering application 34 and in fact rendering with another type of rendering application 34. Assuming the authentication is successful and the digital content 12 is decrypted by the black box 30, the rendering application 34 may then render the decrypted digital content 12 (steps 533, 535).

Sequence of Key Transactions

Referring now to Fig. 10, in one embodiment of the present invention, a sequence of key transactions is performed to obtain the decryption key (KD) and evaluate a license 16 for a requested piece of digital content 12 (i.e., to perform steps 515-523 of Figs. 5A and 5B). Mainly, in such sequence, the DRM system 32 obtains the decryption key (KD) from the license 16, uses information obtained from the license 16 and the digital content 12 to authenticate or ensure the validity of both, and then determines whether the license 16 in fact provides the right to render the digital content 12 in the manner sought. If so, the digital content 12 may be rendered.

Bearing in mind that each license 16 for the digital content 12, as seen in Fig. 8, includes:

- the content ID of the digital content 12 to which the license 16 applies;

- the Digital Rights License (DRL) 48, perhaps encrypted with the decryption key (KD) (i.e., KD (DRL));

- the decryption key (KD) for the digital content 12 encrypted with the black box 30 public key (PU-BB) (i.e.,(PU-BB (KD));

- the digital signature from the license server 24 based on (KD (DRL)) and (PU-BB (KD)) and encrypted with the license server 24 private key (i.e., (S (PR-LS))); and

- the certificate that the license server 24 obtained previously from the content server 22 (i.e., (CERT (PU-LS) S (PR-CS))),

and also bearing in mind that the package 12p having the digital content 12, as seen in Fig. 3, includes:

- the content ID of such digital content 12;

- the digital content 12 encrypted by KD (i.e., (KD(CONTENT)));

- a license acquisition script that is not encrypted; and

- the key KD encrypting the content server 22 public key (PU-CS), signed by the content server 22 private key (PR-CS) (i.e., (KD (PU-CS) S (PR-CS))),

in one embodiment of the present invention. the specific sequence of key transactions that are performed with regard to a specific one of the licenses 16 for the digital content 12 is as follows:

1. Based on (PU-BB (KD)) from the license 16. the black box 30 of the DRM system 32 on the user's computing device 14 applies its private key (PR-

BB) to obtain (KD) (step 1001). (PR-BB (PU-BB (KD)) = (KD)). Note, importantly, that the black box 30 could then proceed to employ KD to decrypt the digital content 12 without any further ado. However, and also importantly, the license server 24 trusts the black box 30 not to do so. Such trust was established at the time such license server 24 issued the license 16 based on the certificate from the certifying authority vouching for the trustworthiness of such black box 30. Accordingly, despite the black box 30 obtaining the decryption key (KD) as an initial step rather than a final step, the DRM system 32 continues to perform all license 16 validation and evaluation functions, as described below.

2. Based on (KD (PU-CS) S (PR-CS)) from the digital content 12, the black box 30 applies the newly obtained decryption key (KD) to obtain (PU-CS) (step 1003). (KD (KD (PU-CS)) = (PU-CS)). Additionally, the black box 30 can apply (PU-CS) as against the signature (S (PR-CS)) to satisfy itself that such signature and such digital content 12 / package 12p is valid (step 1005). If not valid, the process is halted and access to the digital content 12 is denied.

3. Based on (CERT (PU-LS) S (PR-CS)) from the license 16, the black box 30 applies the newly obtained content server 22 public key (PU-CS) to satisfy itself that the certificate is valid (step 1007), signifying that the license server 24 that issued the license 16 had the authority from the content server 22 to do so, and then examines the certificate contents to obtain (PU-LS) (step 1009). If not valid, the process is halted and access to the digital content 12 based on the license 16 is denied.

4. Based on (S (PR-LS)) from the license 16, the black box 30 applies the newly obtained license server 24 public key (PU-LS) to satisfy itself that the license 16 is valid (step 1011). If not valid, the process is halted and access to the digital content 12 based on the license 16 is denied.

5. Assuming all validation steps are successful, and that the DRL 48 in the license 16 is in fact encrypted with the decryption key (KD), the license evaluator 36 then applies the already-obtained decryption key (KD) to (KD(DRL)) as obtained from the license 16 to obtain the license terms from the license 16 (i.e., the

DRL 48) (step 1013). Of course, if the DRL 48 in the license 16 is not in fact encrypted with the decryption key (KD), step 1013 may be omitted. The license evaluator 36 then evaluates / interrogates the DRL 48 and determines whether the user's computing device 14 has the right based on the DRL 48 in the license 16 to render the corresponding digital content 12 in the manner sought (i.e., whether the DRL 48 is enabling) (step 1015). If the license evaluator 36 determines that such right does not exist, the process is halted and access to the digital content 12 based on the license 16 is denied.

6. Finally, assuming evaluation of the license 16 results in a positive determination that the user's computing device 14 has the right based on the DRL 48 terms to render the corresponding digital content 12 in the manner sought, the license evaluator 36 informs the black box 30 that such black box 30 can render the corresponding digital content 12 according to the decryption key (KD). The black box 30 thereafter applies the decryption key (KD) to decrypt the digital content 12 from the package 12p (i.e., (KD(KD(CONTENT)) = (CONTENT)) (step 1017).

It is important to note that the above-specified series of steps represents an alternating or 'ping-ponging' between the license 16 and the digital content 12. Such ping-ponging ensures that the digital content 12 is tightly bound to the license 16, in that the validation and evaluation process can only occur if both the digital content 12 and license 16 are present in a properly issued and valid form. In addition, since the same decryption key (KD) is needed to get the content server 22 public key (PU-CS) from the license 16 and the digital content 12 from the package 12p in a decrypted form (and perhaps the license terms (DRL 48) from the license 16 in a decrypted form), such items are also tightly bound. Signature validation also ensures that the digital content 12 and the license 16 are in the same form as issued from the content server 22 and the license server 24, respectively. Accordingly, it is difficult if not impossible to decrypt the digital content 12 by bypassing the license server 24, and also difficult if not impossible to alter and then decrypt the digital content 12 or the license 16.

In one embodiment of the present invention, signature verification, and especially signature verification of the license 16, is alternately performed as follows. Rather than having a signature encrypted by the private key of the license server 16 (PR-LS), as is seen in Fig. 8, each license 16 has a signature encrypted by a private root key (PR-R) (not shown), where the black box 30 of each DRM system 32 includes a public root key (PU-R) (also not shown) corresponding to the private root key (PR-R). The private root key (PR-R) is known only to a root entity, and a license server 24 can only issue licenses 16 if such license server 24 has arranged with the root entity to issue licenses 16.

In particular, in such embodiment:

1. the license server 24 provides its public key (PU-LS) to the root entity;
2. the root entity returns the license server public key (PU-LS) to such license server 24 encrypted with the private root key (PR-R) (i.e., (CERT (PU-LS) S (PR-R))); and
3. the license server 24 then issues a license 16 with a signature encrypted with the license server private key (S (PR-LS)), and also attaches to the license the certificate from the root entity (CERT (PU-LS) S (PR-R)).

For a DRM system 18 to validate such issued license 16, then, the DRM system 18:

1. applies the public root key (PU-R) to the attached certificate (CERT (PU-LS) S (PR-R)) to obtain the license server public key (PU-LS); and
2. applies the obtained license server public key (PU-LS) to the signature of the license 16 (S (PR-LS).

Importantly, it should be recognized that just as the root entity gave the license server 24 permission to issue licenses 16 by providing the certificate (CERT (PU-LS) S (PR-R)) to such license server 24, such license server 24 can provide a similar certificate to a second license server 24 (i.e., (CERT (PU-LS2) S (PR-LS1)), thereby allowing the second license server to also issue licenses 16. As should now be evident, a license 16 issued by the second license server would include a first certificate (CERT (PU-LS1) S (PR-R)) and a second certificate (CERT (PU-LS2) S (PR-LS1)). Likewise, such license 16 is validated by following the chain through the first and second certificates. Of course, additional links in the chain may be added and traversed.

One advantage of the aforementioned signature verification process is that the root entity may periodically change the private root key (PR-R), thereby likewise periodically requiring each license server 24 to obtain a new certificate (CERT (PU-LS) S (PR-R)). Importantly, as a requirement for obtaining such new certificate, each license server may be required to upgrade itself. As with the black box 30, if a license server 24 is relatively current, i.e., has been upgraded relatively recently, it is less likely that license server 24 has been successfully attacked. Accordingly, as a matter of trust, each license server 24 is preferably required to be upgraded periodically via an appropriate upgrade trigger mechanism such as the signature verification process. Of course, other upgrade mechanisms may be employed without departing from the spirit and scope of the present invention.

Of course, if the private root key (PR-R) is changed, then the public root key (PU-R) in each DRM system 18 must also be changed. Such change may for example take place during a normal black box 30 upgrade, or in fact may require that a black box 30 upgrade take place. Although a changed public root key (PU-R) may potentially interfere with signature validation for an older license 16 issued based on an older private root key (PR-R), such interference may be minimized by requiring that an upgraded black box 30 remember all old public root keys (PU-R). Alternatively, such interference may be minimized by requiring signature verification for a license 16 only once, for example the first time such license 16 is evaluated by the license evaluator 36 of a DRM system 18. In such case, state information on whether signature verification has taken place should be compiled, and such state information should be stored in the state store 40 of the DRM system 18.

Digital Rights License 48

In the present invention, the license evaluator 36 evaluates a Digital Rights License (DRL) 48 as the rights description or terms of a license 16 to determine if such DRL 48 allows rendering of a corresponding piece of digital content 12 in the manner sought. In one embodiment of the present invention, the DRL 48 may be written by a licensor (i.e., the content owner) in any DRL language.

As should be understood, there are a multitude of ways to specify a DRL 48. Accordingly, a high degree of flexibility must be allowed for in any DRL language. However, it is impractical to specify all aspects of a DRL 48 in a particular license language, and it is highly unlikely that the author of such a language can appreciate all possible licensing aspects that a particular digital licensor may desire. Moreover, a highly sophisticated license language may be unnecessary and even a hindrance for a licensor providing a relatively simple DRL 48. Nevertheless, a licensor should not be unnecessarily restricted in how to specify a DRL 48. At the same time, the license evaluator 36 should always be able to get answers from a DRL 48 regarding a number of specific license questions.

In the present invention, and referring now to Fig. 11, a DRL 48 can be specified in any license language, but includes a language identifier or tag 54. The license evaluator 36 evaluating the license 16, then, performs the preliminary step of reviewing the language tag 54 to identify such language, and then selects an appropriate license language engine 52 for accessing the license 16 in such identified language. As should be understood, such license language engine 52 must be present and accessible to the license evaluator 36. If not present, the language tag 54 and/or the DRL 48 preferably includes a location 56 (typically a web site) for obtaining such language engine 52.

Typically, the language engine 52 is in the form of an executable file or set of files that reside in a memory of the user's computing device 14, such as a hard drive. The language engine 52 assists the license evaluator 36 to directly interrogate the DRL 48, the license evaluator 36 interrogates the DRL 48 indirectly via the language engine 48 acting as an intermediary, or the like. When executed, the language engine 52 runs in a work space in a memory of the user's computing device 14, such as RAM. However, any other form of language engine 52 may be employed without departing from the spirit and scope of the present invention.

Preferably, any language engine 52 and any DRL language supports at least a number of specific license questions that the license evaluator 36 expects to be answered by any DRL 48, as will be discussed below. Accordingly, the license evaluator 36 is not tied to any particular DRL language; a DRL 48 may be written in any appropriate DRL language; and a DRL 48 specified in a new license language can be employed by an existing license evaluator 36 by having such license evaluator 36 obtain a corresponding new language engine 52.

DRL Languages

Two examples of DRL languages, as embodied in respective DRLs 48, are provided below. The first, 'simple' DRL 48 is written in a DRL language that specifies license attributes, while the second 'script' DRL 48 is written in a DRL language that can perform functions according to the script specified in the DRL 48. While written in a DRL language, the meaning of each line of code should be apparent based on the linguistics thereof and/or on the attribute description chart that follows:

Simple DRL 48:

```
<LICENSE>
    <DATA>
        <NAME>Beastie Boy's Play</NAME>
        <ID>39384</ID>
        <DESCRIPTION>Play the song 3 times</DESCRIPTION>
        <TERMS></TERMS>
        <VALIDITY>
            <NOTBEFORE>19980102 23:20:14Z</NOTBEFORE>
            <NOTAFTER>19980102 23:20:14Z</NOTAFTER>
        </VALIDITY>
        <ISSUEDDATE>19980102 23:20:14Z</ISSUEDDATE>
        <LICENSORSITE>http://www.foo.com</LICENSORSITE>
```

```
<CONTENT>
    <NAME>Beastie Boy's</NAME>
    <ID>392</ID>
    <KEYID>39292</KEYID>
    <TYPE>MS Encrypted ASF 2.0</TTYPE>
</CONTENT>
<OWNER>
    <ID>939KDKD393KD</ID>
    <NAME>Universal</NAME>
    <PUBLICKEY></PUBLICKEY>
</OWNER>
<LICENSEE>
    <NAME>Arnold</NAME>
    <ID>939KDKD393KD</ID>
    <PUBLICKEY></PUBLICKEY>
</LICENSEE>
<PRINCIPAL TYPE='AND'>
    <PRINCIPAL TYPE='OR'>
        <PRINCIPAL>
            <TYPE>x86Computer</TYPE>
            <ID>3939292939d9e939</ID>
            <NAME>Personal Computer</NAME>
            <AUTHTYPE>Intel Authenticated Boot PC
            SHA-1 DSA512</AUTHTYPE>
            <AUTHDATA>29293939</AUTHDATA>
        </PRINCIPAL>
        <PRINCIPAL>
            <TYPE>Application</TYPE>
            <ID>2939495939292</ID>
            <NAME>Window's Media Player</NAME>
            <AUTHTYPE>Authenticode SHA-
            1</AUTHTYPE>
            <AUTHDATA>93939</AUTHDATA>
        </PRINCIPAL>
    </PRINCIPAL>
    <PRINCIPAL>
        <TYPE>Person</TYPE>
        <ID>39299482010</ID>
        <NAME>Arnold Blinn</NAME>
        <AUTHTYPE>Authenticate user</AUTHTYPE>
        <AUTHDATA>\\redmond\arnoldb</AUTHDATA>
    </PRINCIPAL>
</PRINCIPAL>
```

```
            <DRLTYPE>Simple</DRLTYPE>  [the language tag 54]
            <DRLDATA>
                  <START>19980102 23:20:14Z</START>
                  <END>19980102 23:20:14Z</END>
                  <COUNT>3</COUNT>
                  <ACTION>PLAY</ACTION>
            </DRLDATA>
            <ENABLINGBITS>aaaabbbbccccdddd</ENABLINGBITS>
      </DATA>
      <SIGNATURE>
      <SIGNERNAME>Universal</SIGNERNAME>
            <SIGNERID>9382ABK3939DKD</SIGNERID>
            <HASHALGORITHMID>MD5</HASHALGORITHMID>
            <SIGNALGORITHMID>RSA 128</SIGNALGORITHMID>
            <SIGNATURE>xxxyyyxxxyyyxxxyyy</SIGNATURE>
            <SIGNERPUBLICKEY></SIGNERPUBLICKEY>
            <CONTENTSIGNEDSIGNERPUBLICKEY></CONTENTSIGNEDSI
            GNERPUBLICKEY>
      </SIGNATURE>
</LICENSE>
```


Script DRL 48:


```
<LICENSE>
      <DATA>
            <NAME>Beastie Boy's Play</NAME>
            <ID>39384</ID>
            <DESCRIPTION>Play the song unlimited</DESCRIPTION>
            <TERMS></TERMS>
            <VALIDITY>
                  <NOTBEFORE>19980102 23:20:14Z</NOTBEFORE>
                  <NOTAFTER>19980102 23:20:14Z</NOTAFTER>
            </VALIDITY>
            <ISSUEDDATE>19980102 23:20:14Z</ISSUEDDATE>
            <LICENSORSITE>http://www.foo.com</LICENSORSITE>
            <CONTENT>
                  <NAME>Beastie Boy's</NAME
                  <ID>392</ID>
                  <KEYID>39292</KEYID>
                  <TYPE>MS Encrypted ASF 2.0</TTYPE>
            </CONTENT>
            <OWNER>
                  <ID>939KDKD393KD</ID>
```

```
                <NAME>Universal</NAME>
                <PUBLICKEY></PUBLICKEY>
            </OWNER>
            <LICENSEE>
                <NAME>Arnold</NAME>
                <ID>939KDKD393KD</ID>
                <PUBLICKEY></PUBLICKEY>
            </LICENSEE>
            <DRLTYPE>Script</DRLTYPE>    [the language tag 54]
            <DRLDATA>
                function on_enable(action. args) as boolean
                    result = False
                    if action = "PLAY" then
                        result = True
                    end if
                    on_action = False
                end function
                . . .
            </DRLDATA>
        </DATA>
        <SIGNATURE>
            <SIGNERNAME>Universal</SIGNERNAME>
            <SIGNERID>9382</SIGNERID>
            <SIGNERPUBLICKEY></SIGNERPUBLICKEY>
            <HASHID>MD5</HASHID>
            <SIGNID>RSA 128</SIGNID>
            <SIGNATURE>xxxyyyxxxyyyxxxyyy</SIGNATURE>
            <CONTENTSIGNEDSIGNERPUBLICKEY></CONTENTSIGNEDSI
            GNERPUBLICKEY>
        </SIGNATURE>
</LICENSE>
```

In the two DRLs 48 specified above. the attributes listed have the following descriptions and data types:

Attribute	**Description**	**Data Type**
Id	ID of the license	GUID
Name	Name of the license	String
Content Id	ID of the content	GUID
Content Key Id	ID for the encryption key of the content	GUID
Content Name	Name of the content	String
Content Type	Type of the content	String

Owner Id	ID of the owner of the content	GUID
Owner Name	Name of the owner of the content	String
Owner Public Key	Public key for owner of content. This is a base-64 encoded public key for the owner of the content.	String
Licensee Id	Id of the person getting license. It may be null.	GUID
Licensee Name	Name of the person getting license. It may be null.	String
Licensee Public Key	Public key of the licensee. This is the base-64 encoded public key of the licensee. It may be null.	String
Description	Simple human readable description of the license	String
Terms	Legal terms of the license. This may be a pointer to a web page containing legal prose.	String
Validity Not After	Validity period of license expiration	Date
Validity Not Before	Validity period of license start	Date
Issued Date	Date the license was issued	Date
DRL Type	Type of the DRL. Example include "SIMPLE" or "SCRIPT"	String
DRL Data	Data specific to the DRL	String
Enabling Bits	These are the bits that enable access to the actual content. The interpretation of these bits is up to the application, but typically this will be the private key for decryption of the content. This data will be base-64 encoded. Note that these bits are encrypted using the public key of the individual machine.	String
Signer Id	ID of person signing license	GUID
Signer Name	Name of person signing license	String
Signer Public Key	Public key for person signing license. This is the base-64 encode public key for the signer.	String
Content Signed Signer Public Key	Public key for person signing the license that has been signed by the content server private key. The public key to verify this signature will be encrypted in the content. This is base-64 encoded.	String

Hash Alg Id	Algorithm used to generate hash. This is a string, such as "MD5".	String
Signature Alg Id	Algorithm used to generate signature. This is a string, such as "RSA 128".	String
Signature	Signature of the data. This is base-64 encoded data.	String

Methods

As was discussed above, it is preferable that any language engine 52 and any DRL language support at least a number of specific license questions that the digital license evaluator 36 expects to be answered by any DRL 48. Recognizing such supported questions may include any questions without departing from the spirit and scope of the present invention, and consistent with the terminology employed in the two DRL 48 examples above, in one embodiment of the present invention, such supported questions or 'methods' include 'access methods', 'DRL methods', and 'enabling use methods', as follows:

Access Methods

Access methods are used to query a DRL 48 for top-level attributes.

VARIANT QueryAttribute (BSTR key)

Valid keys include License.Name, License.Id, Content.Name, Content.Id, Content.Type, Owner.Name, Owner.Id, Owner.PublicKey, Licensee.Name, Licensee.Id, Licensee.PublicKey, Description, and Terms, each returning a BSTR variant; and Issued, Validity.Start and Validity.End, each returning a Date Variant.

DRL Methods

The implementation of the following DRL methods varies from DRL 48 to DRL 48. Many of the DRL methods contain a variant parameter labeled 'data' which is intended for communicating more advanced information with a DRL 48. It is present largely for future expandability.

Boolean IsActivated(Variant data)

This method returns a Boolean indicating whether the DRL 48 / license 16 is activated. An example of an activated license 16 is a limited operation license 16 that upon first play is active for only 48 hours.

Activate(Variant data)

This method is used to activate a license 16. Once a license 16 is activated, it cannot be deactivated.

Variant QueryDRL(Variant data)

This method is used to communicate with a more advanced DRL 48. It is largely about future expandability of the DRL 48 feature set.

Variant GetExpires(BSTR action, Variant data)

This method returns the expiration date of a license 16 with regard to the passed-in action. If the return value is NULL, the license 16 is assumed to never expire or does not yet have an expiration date because it hasn't been activated. or the like.

Variant GetCount(BSTR action, Variant data)

This method returns the number of operations of the passed-in action that are left. If NULL is returned, the operation can be performed an unlimited number of times.

Boolean IsEnabled(BSTR action, Variant data)

This method indicates whether the license 16 supports the requested action at the present time.

Boolean IsSunk(BSTR action, Variant data)

This method indicates whether the license 16 has been paid for. A license 16 that is paid for up front would return TRUE, while a license 16 that is not paid for up front, such as a license 16 that collects payments as it is used, would return FALSE.

Enabling Use Methods.

These methods are employed to enable a license 16 for use in decrypting content.

Boolean Validate (BSTR key)

This method is used to validate a license 16. The passed-in key is the black box 30 public key (PU-BB) encrypted by the decryption key (KD) for the corresponding digital content 12 (i.e.,(KD(PU-BB))) for use in validation of the signature of the license 16. A return value of TRUE indicates that the license 16 is valid. A return value of FALSE indicates invalid.

int OpenLicense 16(BSTR action, BSTR key. Variant data)

This method is used to get ready to access the decrypted enabling bits. The passed-in key is (KD(PU-BB)) as described above. A return value of 0 indicates success. Other return values can be defined.

BSTR GetDecryptedEnablingBits (BSTR action. Variant data)

Variant GetDecryptedEnablingBitsAsBinary (BSTR action. Variant Data)

These methods are used to access the enabling bits in decrypted form. If this is not successful for any of a number of reasons, a null string or null variant is returned.

void CloseLicense 16 (BSTR action, Variant data)

This method is used to unlock access to the enabling bits for performing the passed-in action. If this is not successful for any of a number of reasons. a null string is returned.

Heuristics

As was discussed above, if multiple licenses 16 are present for the same piece of digital content 12, one of the licenses 16 must be chosen for further use. Using the above methods, the following heuristics could be implemented to make such choice. In particular, to perform an action (say "PLAY") on a piece of digital content 12, the following steps could be performed:

1. Get all licenses 16 that apply to the particular piece of digital content 12.
2. Eliminate each license 16 that does not enable the action by calling the IsEnabled function on such license 16.
3. Eliminate each license 16 that is not active by calling IsActivated on such license 16.
4. Eliminate each license 16 that is not paid for up front by calling IsSunk on such license 16.
5. If any license 16 is left, use it. Use an unlimited-number-of-plays license 16 before using a limited-number-of-plays license 16, especially if the unlimited-number-of-plays license 16 has an expiration date. At any time, the user should be allowed to select a specific license 16 that has already been acquired. even if the choice is not cost-effective. Accordingly, the user can select a license 16 based on criteria that are perhaps not apparent to the DRM system 32.
6. If there are no licenses 16 left. return status so indicating. The user would then be given the option of:

   using a license 16 that is not paid for up front. if available;
   activating a license 16. if available; and/or
   performing license acquisition from a license server 24.

CONCLUSION

The programming necessary to effectuate the processes performed in connection with the present invention is relatively straight-forward and should be apparent to the relevant programming public. Accordingly, such programming is not attached hereto. Any particular programming, then, may be employed to effectuate the present invention without departing from the spirit and scope thereof.

In the foregoing description, it can be seen that the present invention comprises a new and useful enforcement architecture 10 that allows the controlled rendering or playing of arbitrary forms of digital content 12, where such control is flexible and definable by the content owner of such digital content 12. Also, the present invention comprises a new useful controlled rendering environment that renders digital content 12 only as specified by the content owner, even though the digital content 12 is to be rendered on a computing device 14 which is not under the control of the content owner. Further, the present invention comprises a trusted component that enforces the rights of the content owner on such computing device 14 in connection with a piece of digital content 12, even against attempts by the user of such computing device 14 to access such digital content 12 in ways not permitted by the content owner.

It should be appreciated that changes could be made to the embodiments described above without departing from the inventive concepts thereof. It should be understood, therefore, that this invention is not limited to the particular embodiments disclosed, but it is intended to cover modifications within the spirit and scope of the present invention as defined by the appended claims.

CLAIMS

1. A method for a device to interdependently validate:

a digital content package having a piece of digital content in an encrypted form; and

a corresponding digital license for rendering the digital content,

the method comprising:

deriving a first key from a source available to the device;

obtaining a first digital signature from the digital content package;

applying the first key to the first digital signature to validate the first digital signature and the digital content package;

deriving a second key based on the first digital signature;

obtaining a second digital signature from the license; and

applying the second key to the second digital signature to validate the second digital signature and the license.

2. The method of claim 1 wherein deriving the first key comprises:

obtaining a first encrypted key from the license;

applying a key available to the device to the first encrypted key to decrypt the first encrypted key;

obtaining a second encrypted key from the digital content; and

applying the decrypted first encrypted key to the second encrypted key to produce the first key.

3. The method of claim 2 wherein the encrypted digital content is decryptable according to a decryption key (KD), and wherein the first encrypted key is the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))).

4. The method of claim 2 wherein the device has a public key (PU-D) and a private key (PR-D), and wherein the key available to the device is (PR-D).

5. The method of claim 2 wherein the encrypted digital content is decryptable according to a decryption key (KD), wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein the second encrypted key is the content provider public key (PU-C) encrypted with the decryption key (KD) (i.e., KD (PU-C)).

6. The method of claim 2 wherein the second encrypted key is the basis for the first digital signature.

7. The method of claim 1 wherein deriving the second key comprises:

obtaining a signed certificate from the license. the signed certificate having contents therein; and

applying the first key to the signature of the signed certificate to produce the contents of the certificate and also to validate the signature.

8. The method of claim 7 wherein the digital license is provided by a license provider having a public key (PU-L) and a private key (PR-L), and wherein the contents of the certificate is (PU-L).

9. The method of claim 8 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein the signed certificate is a certificate containing the license provider public key (PU-L) and signed by the content provider private key (PR-C) (i.e., (CERT (PU-L) S (PR-C))).

10. The method of claim 8 wherein the digital content package is provided by a content provider authorized by a root source to provide the package, wherein the root source has a public key (PU-R) and a private key (PR-R) and wherein the signed certificate is a certificate containing the license provider public key (PU-L) and signed by the root source private key (PR-R) (i.e., (CERT (PU-L) S (PR-R))).

11. The method of claim 1 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein the first key is (PU-C).

12. The method of claim 11 wherein the encrypted digital content is decryptable according to a decryption key (KD), and wherein the first digital signature is based on the content provider public key (PU-C) encrypted with the decryption key (KD) and

is signed by the content provider private key (PR-C) (i.e.. (KD (PU-C) S (PR-C))).

13. The method of claim 12 wherein deriving (PU-C) comprises:

deriving (KD) from a source available to the device;

applying (KD) to (KD (PU-C) S (PR-C)) to produce (PU-C).

14. The method of claim 13 wherein the device has a public key (PU-D) and a private key (PR-D), wherein the license has the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))). and wherein deriving (KD) comprises:

obtaining (PU-D (KD)) from the license;

applying (PR-D) to (PU-D (KD)) to produce (KD).

15 The method of claim 14 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the license rights description being encrypted with the decryption key (KD) (i.e., (KD (DRL))), the method further comprising applying (KD) to (KD(DRL)) to obtain the license terms and conditions.

16. The method of claim 14 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the method further comprising:

evaluating the license terms and conditions to determine whether the digital content is permitted to be rendered in the manner sought;

if so, applying (KD) to the encrypted digital content to decrypt such encrypted digital content; and

rendering the decrypted digital content.

17. The method of claim 11 wherein the encrypted digital content package is provided by a content provider authorized by a root source to provide the package, wherein the root source has a public key (PU-R) and a private key (PR-R) and wherein the first digital signature is a signed certificate containing the content provider public key (PU-C) and signed by the root source private key (PR-R) (i.e., (CERT (PU-C) S (PR-R))).

18. The method of claim 1 wherein the digital license is provided by a license provider having a public key (PU-L) and a private key (PR-L). and wherein the second key is (PU-L).

19. The method of claim 18 wherein the second digital signature is a digital signature encrypted with the license provider private key (i.e.. (S (PR-L))).

20. The method of claim 19 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), wherein the

license has a certificate containing the license provider public key (PU-L) and signed by the content provider private key (PR-C) (i.e., (CERT (PU-L) S (PR-C))), and wherein deriving (PU-L) comprises:

deriving (PU-C) from a source available to the device;

obtaining (CERT (PU-L) S (PR-C)) from the license; and

applying (PU-C) to (CERT (PU-L) S (PR-C)) to validate (CERT (PU-L) S (PR-C)), to produce (PU-L) and also to validate the content provider.

21. The method of claim 20 wherein the encrypted digital content is decryptable according to a decryption key (KD), wherein the first digital signature is based on the content provider public key (PU-C) encrypted with the decryption key (KD) and is signed by the content provider private key (PR-C) (i.e., (KD (PU-C) S (PR-C))), and wherein deriving (PU-C) comprises:

deriving (KD) from a source available to the device;

applying (KD) to (KD (PU-C) S (PR-C)) to produce (PU-C).

22. The method of claim 21 wherein the device has a public key (PU-D) and a private key (PR-D), wherein the license has the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))), and wherein deriving (KD) comprises:

obtaining (PU-D (KD)) from the license;

applying (PR-D) to (PU-D (KD)) to produce (KD).

23. The method of claim 22 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the license rights description being encrypted with the decryption key (KD) (i.e., (KD (DRL))), the method further comprising applying (KD) to (KD(DRL)) to obtain the license terms and conditions.

24. The method of claim 22 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the method further comprising:

evaluating the license terms and conditions to determine whether the digital content is permitted to be rendered in the manner sought;

if so, applying (KD) to the encrypted digital content to decrypt such encrypted digital content; and

rendering the decrypted digital content.

25. A method for a device to interdependently validate a piece of digital content and a corresponding digital license for rendering the digital content. the digital content being encrypted, the encrypted digital content being decryptable according to a decryption key (KD) and being packaged in a digital content package. the digital content package being provided by a content provider having a public key (PU-C) and a private key (PR-C), the digital license being provided by a license provider having

a public key (PU-L) and a private key (PR-L), the device having a public key (PU-D) and a private key (PR-D), the digital content package comprising:

the encrypted digital content; and

the content provider public key (PU-C) encrypted with the decryption key (KD) and signed by the content provider private key (PR-C) (i.e., (KD (PU-C) S (PR-C)));

the digital license comprising:

the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD)));

a digital signature from the license provider (without any attached certificate) based on (KD (DRL)) and (PU-D (KD)) and encrypted with the license provider private key (i.e., (S (PR-L))); and

a certificate containing the license provider public key (PU-L) and signed by the content provider private key (PR-C) (i.e., (CERT (PU-L) S (PR-C)));

the method comprising:

obtaining (PU-D (KD)) from the license;

applying (PR-D) to (PU-D (KD)) to produce (KD);

obtaining (KD (PU-C) S (PR-C)) from the digital content package;

applying (KD) to (KD (PU-C) S (PR-C)) to produce (PU-C);

applying (PU-C) to (S (PR-C)) to validate (KD (PU-C) S (PR-C)), thereby validating the digital content package;

obtaining (CERT (PU-L) S (PR-C)) from the license;

applying (PU-C) to (CERT (PU-L) S (PR-C)) to validate (CERT (PU-L) S (PR-C)), thereby validating the content provider, and also to obtain (PU-L);

obtaining (S (PR-L)) from the license; and

applying (PU-L) to (S (PR-L)), thereby validating the license.

26. The method of claim 25 wherein the digital content package further comprises a content / package ID identifying one of the digital content and the digital content package, and wherein the license further comprises the content / package ID of the corresponding digital content / digital content package, the method further comprising ensuring that the content / package ID of the license in fact corresponds to the content / package ID of the digital content / digital content package.

27. The method of claim 25 wherein the license further comprises a license rights description (DRL) specifying terms and conditions that must be satisfied before the digital content may be rendered, the method further comprising;

evaluating the license terms and conditions to determine whether the digital content is permitted to be rendered in the manner sought;

if so, applying (KD) to the encrypted digital content to decrypt such encrypted digital content; and

rendering the decrypted digital content.

28. The method of claim 27 wherein the license rights description is encrypted with the decryption key (KD) (i.e., (KD (DRL))), the method further comprising applying (KD) to (KD (DRL)) to obtain the license terms and conditions.

29. A computer-readable medium having computer-executable instructions for performing a method for a device to interdependently validate:

a digital content package having a piece of digital content in an encrypted form; and

a corresponding digital license for rendering the digital content, the method comprising:

deriving a first key from a source available to the device;

obtaining a first digital signature from the digital content package;

applying the first key to the first digital signature to validate the first digital signature and the digital content package;

deriving a second key based on the first digital signature;

obtaining a second digital signature from the license; and

applying the second key to the second digital signature to validate the second digital signature and the license.

30. The method of claim 28 wherein deriving the first key comprises:

obtaining a first encrypted key from the license;

applying a key available to the device to the first encrypted key to decrypt the first encrypted key;

obtaining a second encrypted key from the digital content; and

applying the decrypted first encrypted key to the second encrypted key to produce the first key.

31. The method of claim 30 wherein the encrypted digital content is decryptable according to a decryption key (KD), and wherein the first encrypted key is the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))).

32. The method of claim 30 wherein the device has a public key (PU-D) and a private key (PR-D), and wherein the key available to the device is (PR-D).

33. The method of claim 30 wherein the encrypted digital content is decryptable according to a decryption key (KD), wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein the second encrypted key is the content provider public key (PU-C) encrypted with the decryption key (KD) (i.e., KD (PU-C)).

34. The method of claim 30 wherein the second encrypted key is the basis for the first digital signature.

35. The method of claim 29 wherein deriving the second key comprises:

obtaining a signed certificate from the license. the signed certificate having contents therein; and

applying the first key to the signature of the signed certificate to produce the contents of the certificate and also to validate the signature.

36. The method of claim 35 wherein the digital license is provided by a license provider having a public key (PU-L) and a private key (PR-L), and wherein the contents of the certificate is (PU-L).

37. The method of claim 36 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein the signed certificate is a certificate containing the license provider public key (PU-L) and signed by the content provider private key (PR-C) (i.e.. (CERT (PU-L) S (PR-C))).

38. The method of claim 36 wherein the digital content package is provided by a content provider authorized by a root source to provide the package. wherein the root source has a public key (PU-R) and a private key (PR-R) and wherein the signed certificate is a certificate containing the license provider public key (PU-L) and signed by the root source private key (PR-R) (i.e.. (CERT (PU-L) S (PR-R))).

39. The method of claim 29 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), and wherein

the first key is (PU-C).

40. The method of claim 39 wherein the encrypted digital content is decryptable according to a decryption key (KD), and wherein the first digital signature is based on the content provider public key (PU-C) encrypted with the decryption key (KD) and is signed by the content provider private key (PR-C) (i.e., (KD (PU-C) S (PR-C))).

41. The method of claim 40 wherein deriving (PU-C) comprises:

deriving (KD) from a source available to the device;

applying (KD) to (KD (PU-C) S (PR-C)) to produce (PU-C).

42. The method of claim 41 wherein the device has a public key (PU-D) and a private key (PR-D), wherein the license has the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))), and wherein deriving (KD) comprises:

obtaining (PU-D (KD)) from the license;

applying (PR-D) to (PU-D (KD)) to produce (KD).

43. The method of claim 42 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the license rights description being encrypted with the decryption key (KD) (i.e., (KD (DRL))), the method further comprising applying (KD) to (KD(DRL))

to obtain the license terms and conditions.

44. The method of claim 42 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the method further comprising:

evaluating the license terms and conditions to determine whether the digital content is permitted to be rendered in the manner sought;

if so, applying (KD) to the encrypted digital content to decrypt such encrypted digital content; and

rendering the decrypted digital content.

45. The method of claim 39 wherein the encrypted digital content package is provided by a content provider authorized by a root source to provide the package, wherein the root source has a public key (PU-R) and a private key (PR-R) and wherein the first digital signature is a signed certificate containing the content provider public key (PU-C) and signed by the root source private key (PR-R) (i.e., (CERT (PU-C) S (PR-R))).

46. The method of claim 29 wherein the digital license is provided by a license provider having a public key (PU-L) and a private key (PR-L), and wherein the second key is (PU-L).

47. The method of claim 46 wherein the second digital signature is a digital signature encrypted with the license provider private key (i.e., (S (PR-L))).

48. The method of claim 47 wherein the digital content package is provided by a content provider having a public key (PU-C) and a private key (PR-C), wherein the license has a certificate containing the license provider public key (PU-L) and signed by the content provider private key (PR-C) (i.e., (CERT (PU-L) S (PR-C))), and wherein deriving (PU-L) comprises:

deriving (PU-C) from a source available to the device;

obtaining (CERT (PU-L) S (PR-C)) from the license; and

applying (PU-C) to (CERT (PU-L) S (PR-C)) to validate (CERT (PU-L) S (PR-C)), to produce (PU-L) and also to validate the content provider.

49. The method of claim 48 wherein the encrypted digital content is decryptable according to a decryption key (KD), wherein the first digital signature is based on the content provider public key (PU-C) encrypted with the decryption key (KD) and is signed by the content provider private key (PR-C) (i.e.. (KD (PU-C) S (PR-C))), and wherein deriving (PU-C) comprises:

deriving (KD) from a source available to the device:

applying (KD) to (KD (PU-C) S (PR-C)) to produce (PU-C).

50. The method of claim 49 wherein the device has a public key (PU-D) and a

private key (PR-D), wherein the license has the decryption key (KD) encrypted with the device public key (PU-D) (i.e.,(PU-D (KD))), and wherein deriving (KD) comprises:

obtaining (PU-D (KD)) from the license;

applying (PR-D) to (PU-D (KD)) to produce (KD).

51. The method of claim 50 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the license rights description being encrypted with the decryption key (KD) (i.e., (KD (DRL))), the method further comprising applying (KD) to (KD(DRL)) to obtain the license terms and conditions.

52. The method of claim 50 wherein the license has a license rights description specifying terms and conditions that must be satisfied before the digital content may be rendered, the method further comprising:

evaluating the license terms and conditions to determine whether the digital content is permitted to be rendered in the manner sought;

if so, applying (KD) to the encrypted digital content to decrypt such encrypted digital content; and

rendering the decrypted digital content.

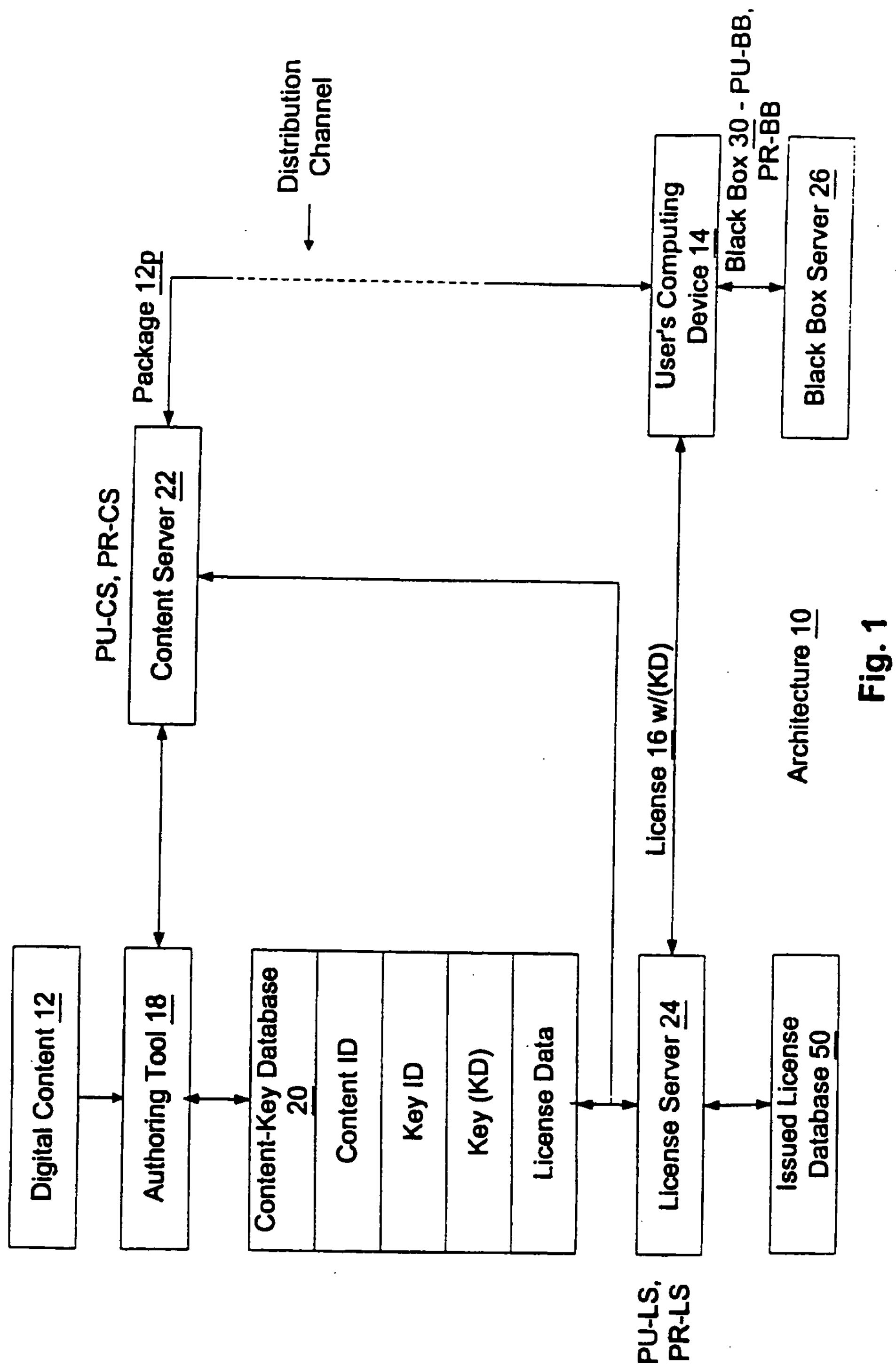

Digital Content 12
Authoring Tool 18
Content-Key Database 20
Content ID
Key ID
Key (KD)
License Data
License Server 24
PU-LS, PR-LS
Issued License Database 50
Content Server 22
PU-CS, PR-CS
Package 12p
Distribution Channel
User's Computing Device 14
Black Box 30 - PU-BB, PR-BB
Black Box Server 26
License 16 w/(KD)
Architecture 10

Fig. 1

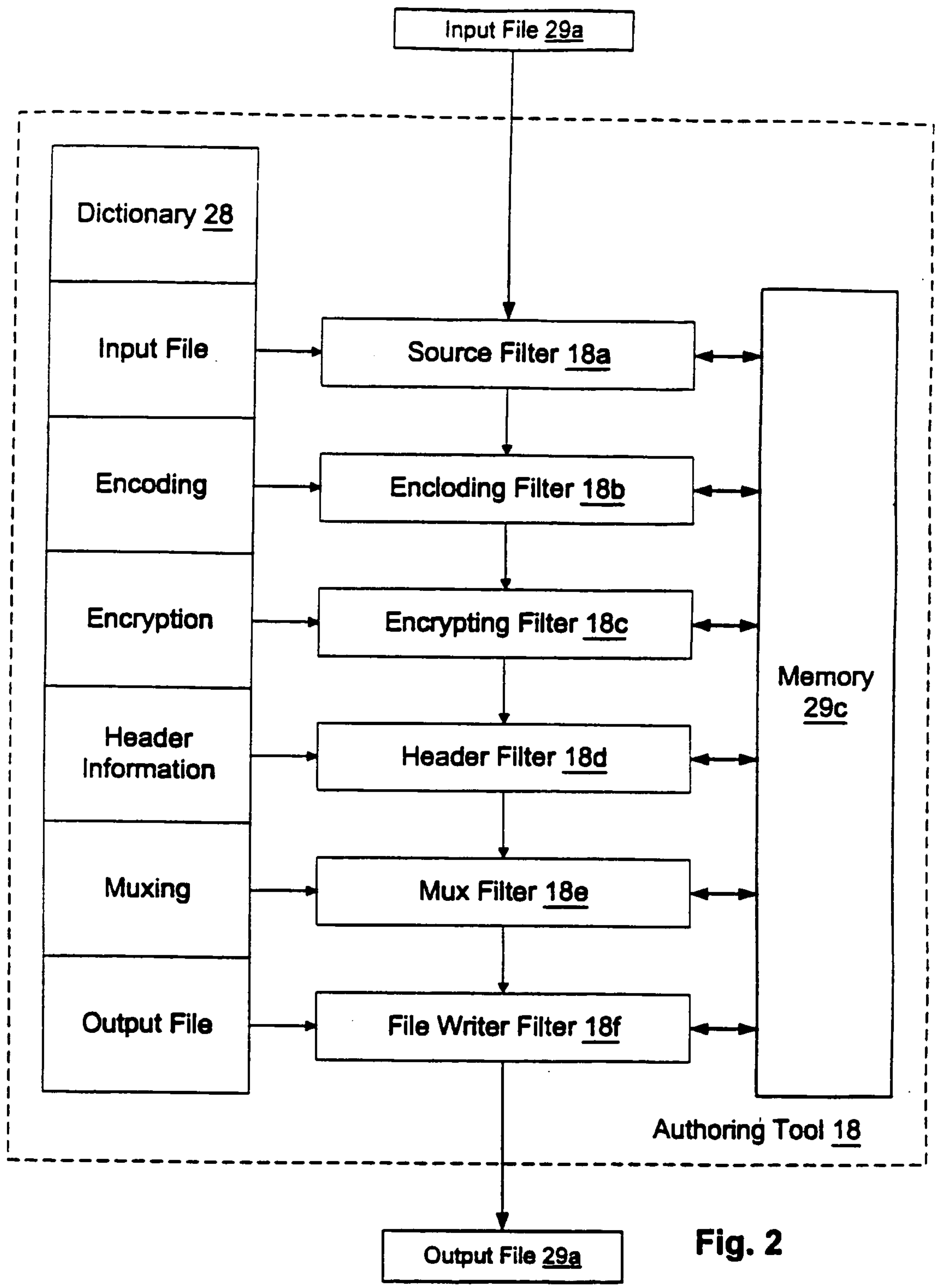

Input File 29a
Dictionary 28
Input File
Encoding
Encryption
Header Information
Muxing
Output File
Source Filter 18a
Enclosing Filter 18b
Encrypting Filter 18c
Header Filter 18d
Mux Filter 18e
File Writer Filter 18f
Memory 29c
Authoring Tool 18
Output File 29a

Fig. 2

Digital Content Package 12p
KD (Digital Content 12)
Content ID
Key ID
License Acquisition Info
KD (PU-CS) S (PR-CS)

Fig. 3

License 16
Content ID
DRL 48 or KD (DRL 48)
PU-BB (KD)
S (PR-LS)
CERT (PU-LS) S (PR-CS)

Fig. 8

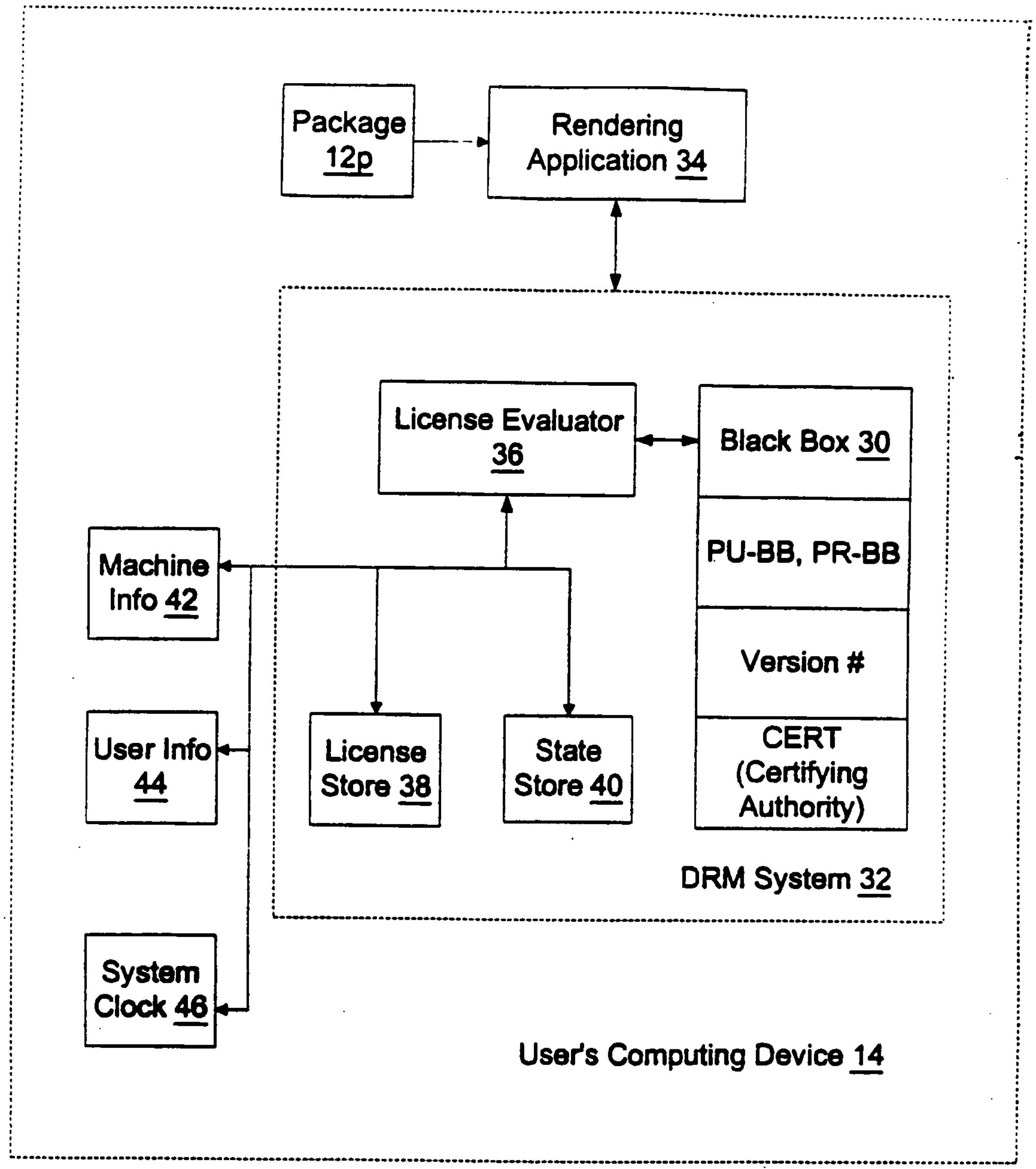

Package 12p
Rendering Application 34
License Evaluator 36
Black Box 30
PU-BB, PR-BB
Version #
CERT (Certifying Authority)
Machine Info 42
User Info 44
License Store 38
State Store 40
DRM System 32
System Clock 46
User's Computing Device 14

Fig. 4

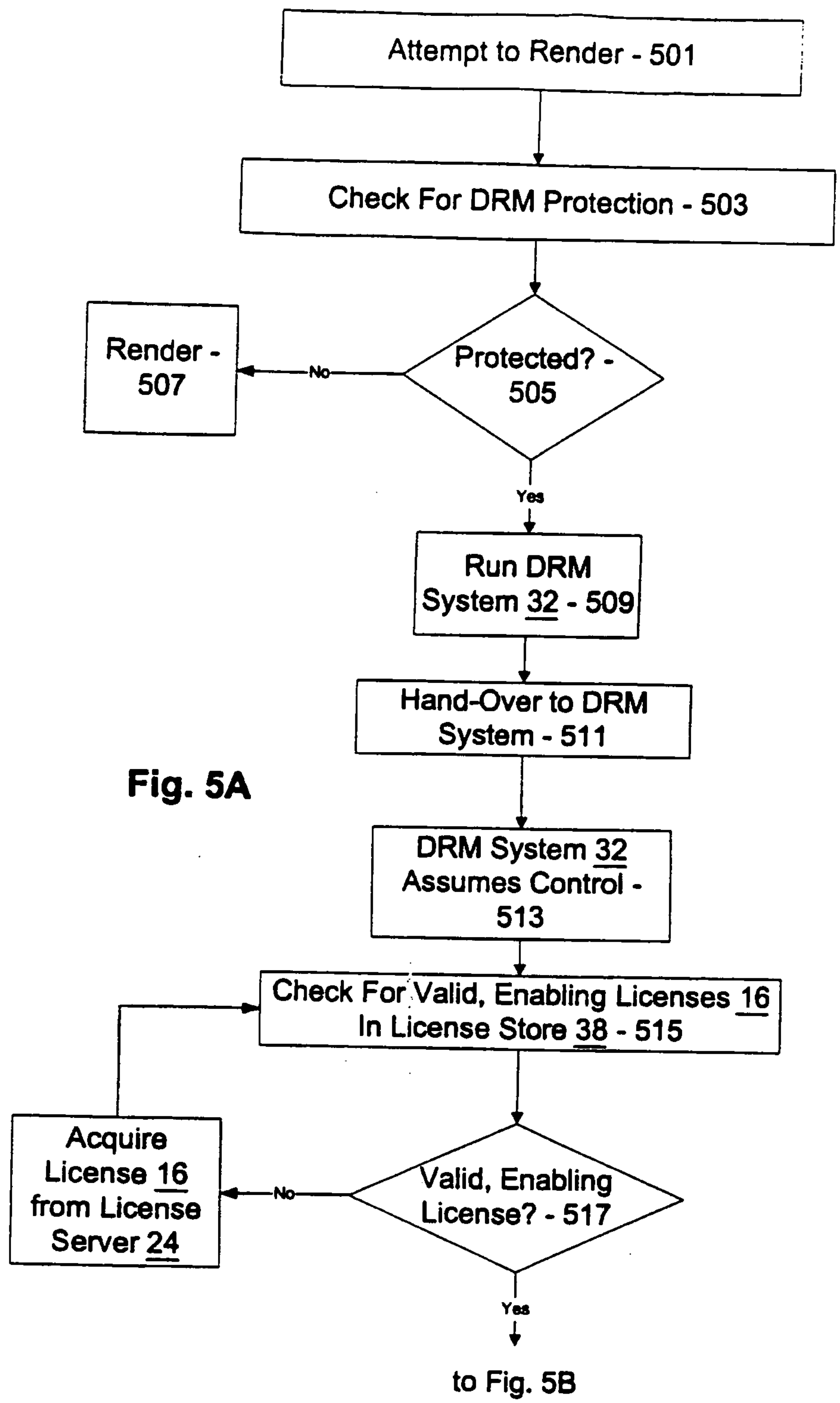

Attempt to Render - 501
Check For DRM Protection - 503
Render - 507
No
Protected? - 505
Yes
Run DRM System 32 - 509
Hand-Over to DRM System - 511
DRM System 32 Assumes Control - 513
Check For Valid, Enabling Licenses 16 In License Store 38 - 515
Acquire License 16 from License Server 24
No
Valid, Enabling License? - 517
Yes
to Fig. 5B

Fig. 5A

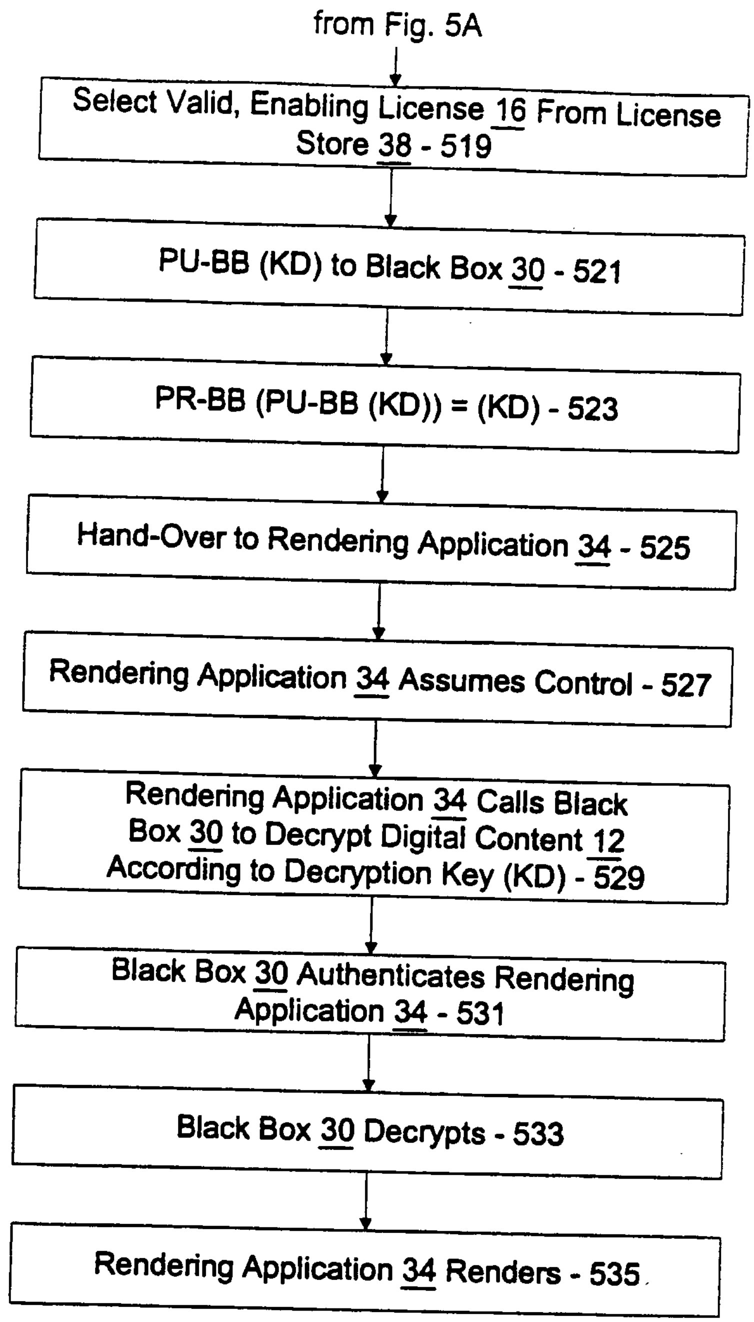

from Fig. 5A
Select Valid, Enabling License 16 From License Store 38 - 519
PU-BB (KD) to Black Box 30 - 521
PR-BB (PU-BB (KD)) = (KD) - 523
Hand-Over to Rendering Application 34 - 525
Rendering Application 34 Assumes Control - 527
Rendering Application 34 Calls Black Box 30 to Decrypt Digital Content 12 According to Decryption Key (KD) - 529
Black Box 30 Authenticates Rendering Application 34 - 531
Black Box 30 Decrypts - 533
Rendering Application 34 Renders - 535

Fig. 5B

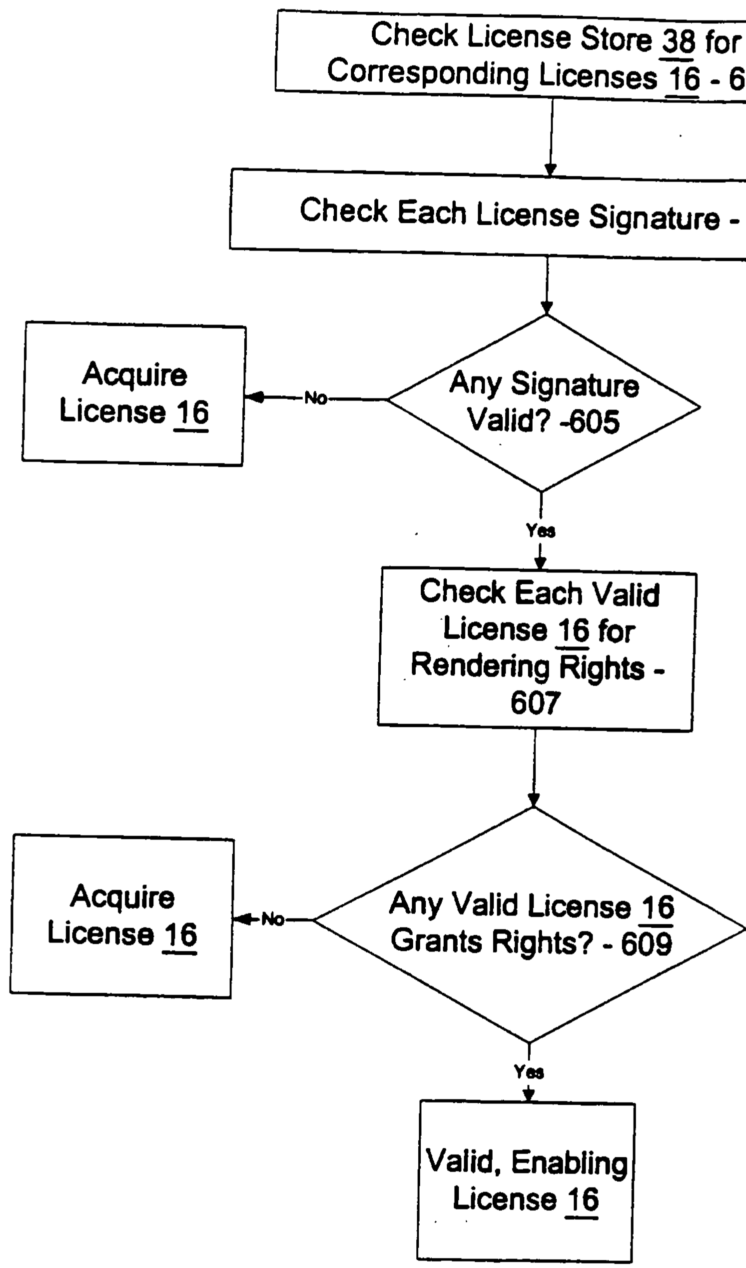

Check License Store 38 for Corresponding Licenses 16 - 601
Check Each License Signature - 603
Any Signature Valid? -605
No
Acquire License 16
Yes
Check Each Valid License 16 for Rendering Rights - 607
Any Valid License 16 Grants Rights? - 609
No
Acquire License 16
Yes
Valid, Enabling License 16

Fig. 6

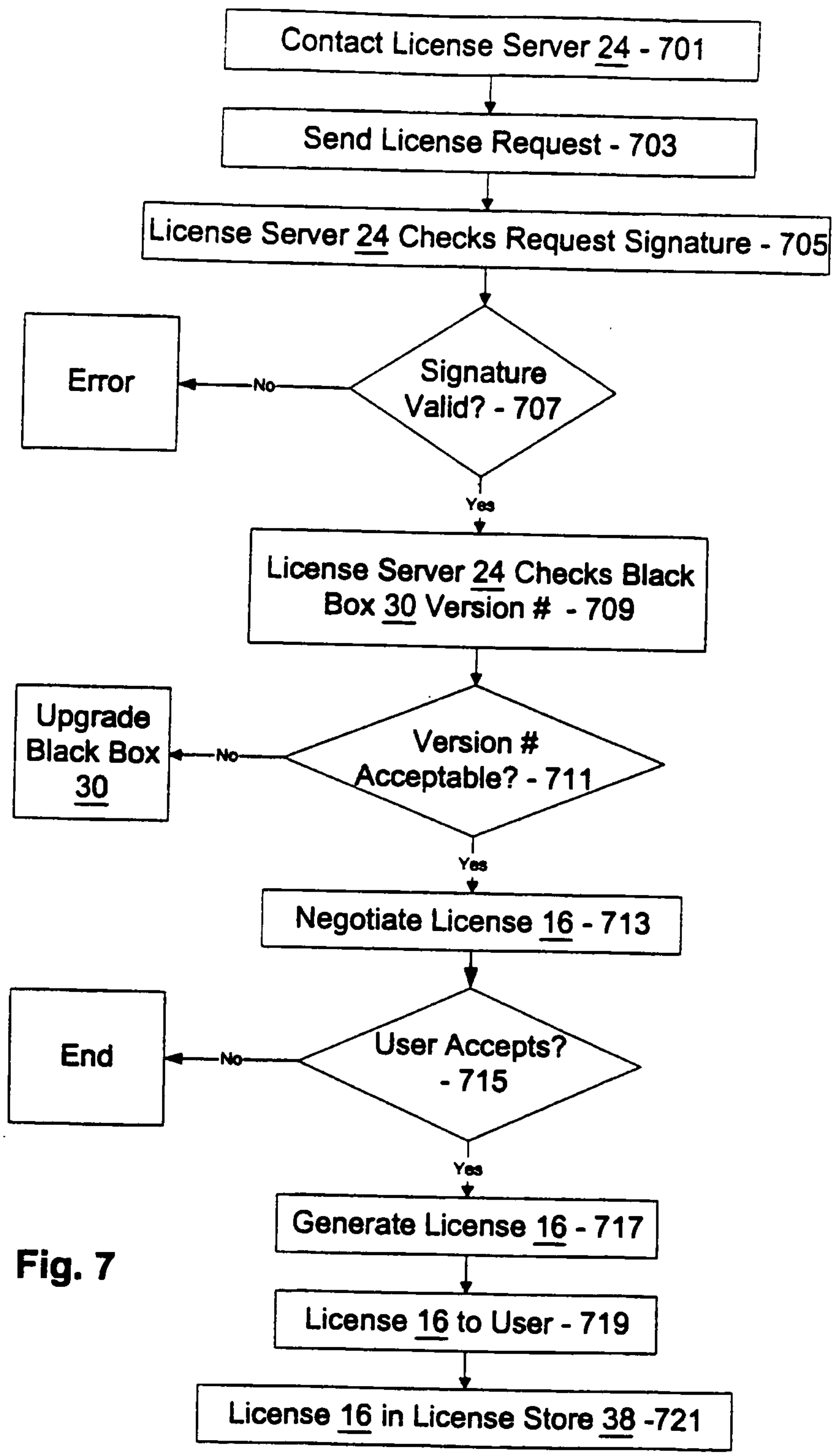

Contact License Server 24 - 701
Send License Request - 703
License Server 24 Checks Request Signature - 705
Signature Valid? - 707
No
Error
Yes
License Server 24 Checks Black Box 30 Version # - 709
Version # Acceptable? - 711
No
Upgrade Black Box 30
Yes
Negotiate License 16 - 713
User Accepts? - 715
No
End
Yes
Generate License 16 - 717
License 16 to User - 719
License 16 in License Store 38 -721

Fig. 7

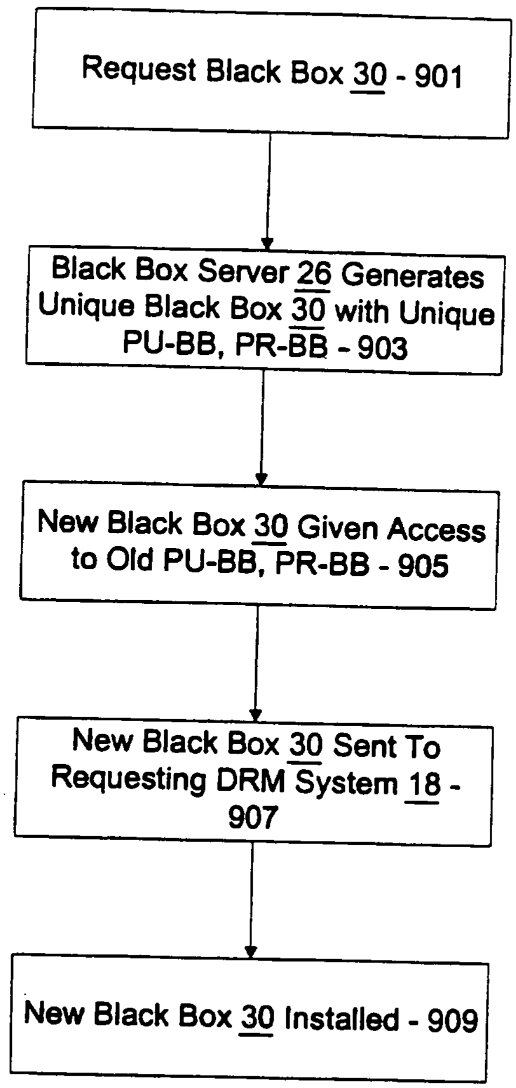

Request Black Box 30 - 901
Black Box Server 26 Generates Unique Black Box 30 with Unique PU-BB, PR-BB - 903
New Black Box 30 Given Access to Old PU-BB, PR-BB - 905
New Black Box 30 Sent To Requesting DRM System 18 - 907
New Black Box 30 Installed - 909

Fig. 9

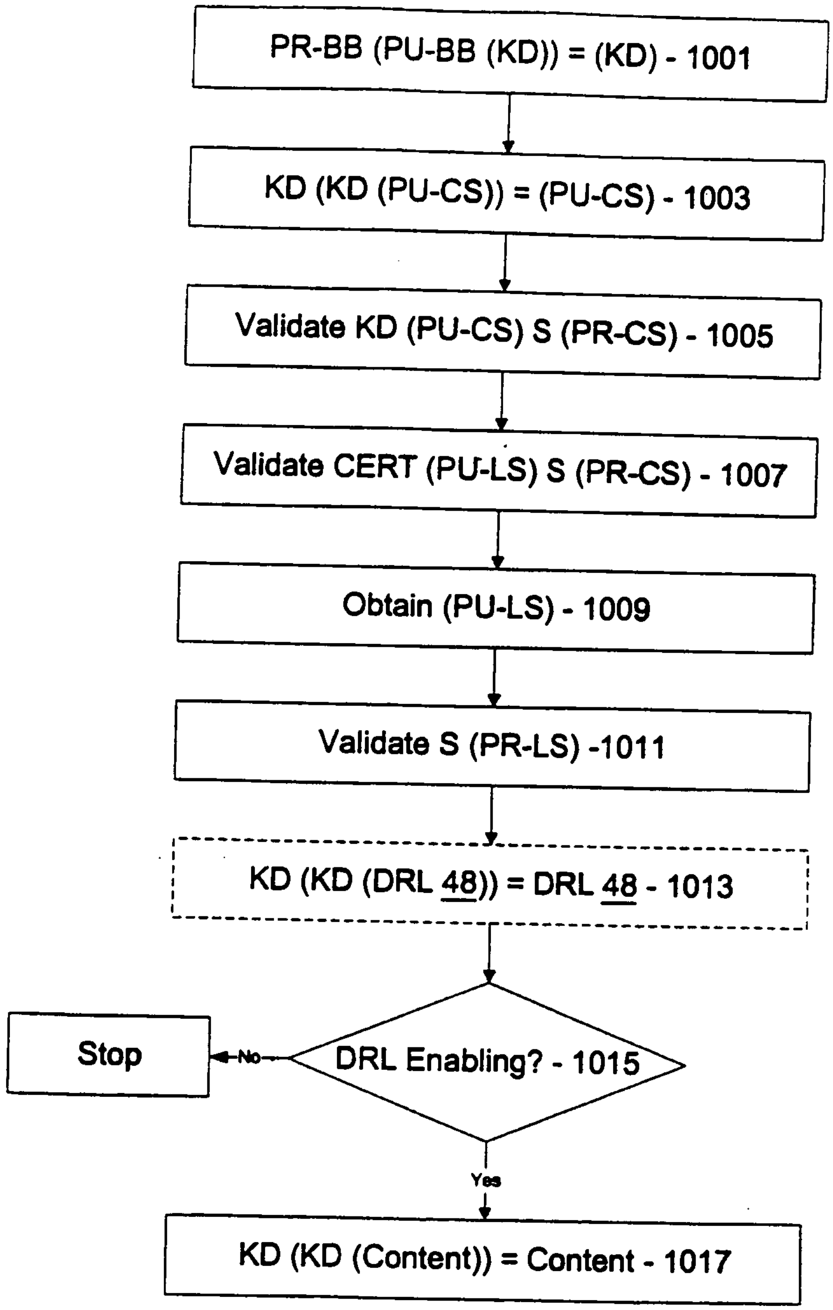

PR-BB (PU-BB (KD)) = (KD) - 1001
KD (KD (PU-CS)) = (PU-CS) - 1003
Validate KD (PU-CS) S (PR-CS) - 1005
Validate CERT (PU-LS) S (PR-CS) - 1007
Obtain (PU-LS) - 1009
Validate S (PR-LS) -1011
KD (KD (DRL 48)) = DRL 48 - 1013
DRL Enabling? - 1015
No
Stop
Yes
KD (KD (Content)) = Content - 1017

Fig. 10

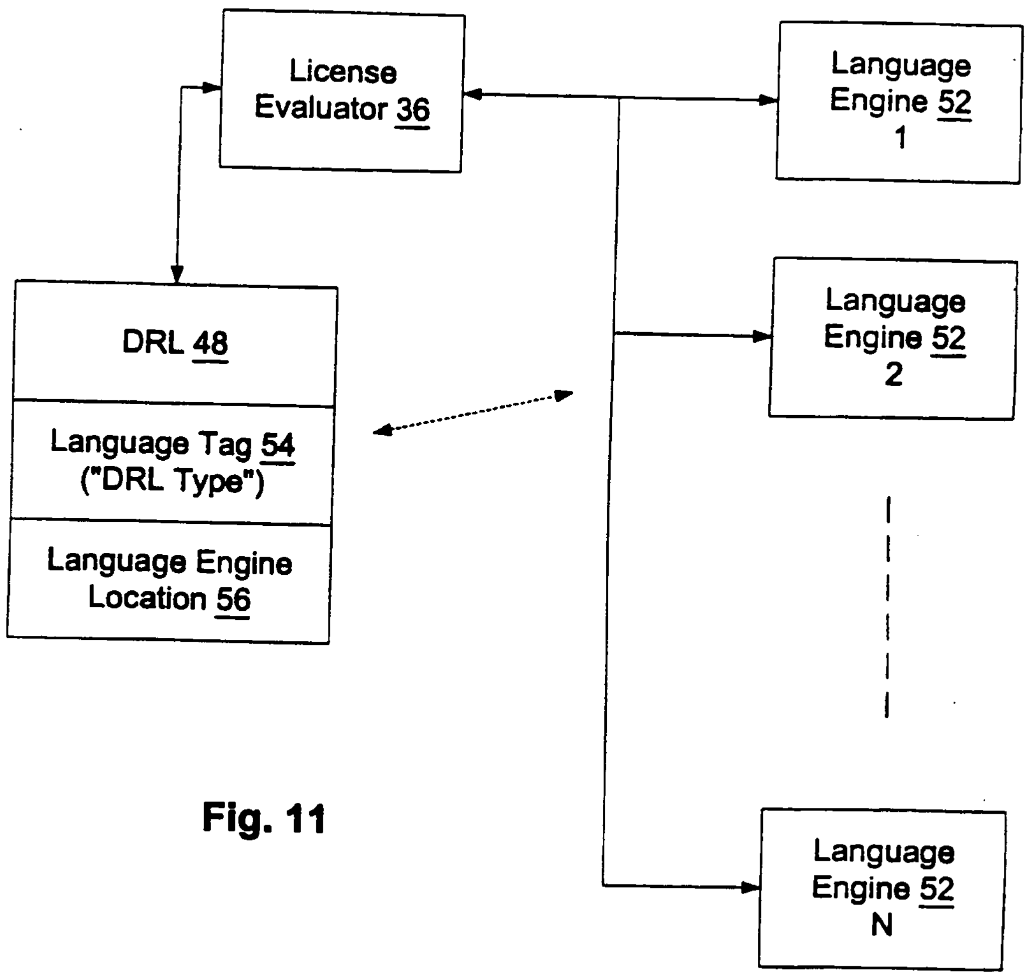

License
Evaluator 36
Language
Engine 52
1
DRL 48
Language
Engine 52
2
Language Tag 54
("DRL Type")
Language Engine
Location 56
Language
Engine 52
N

Fig. 11

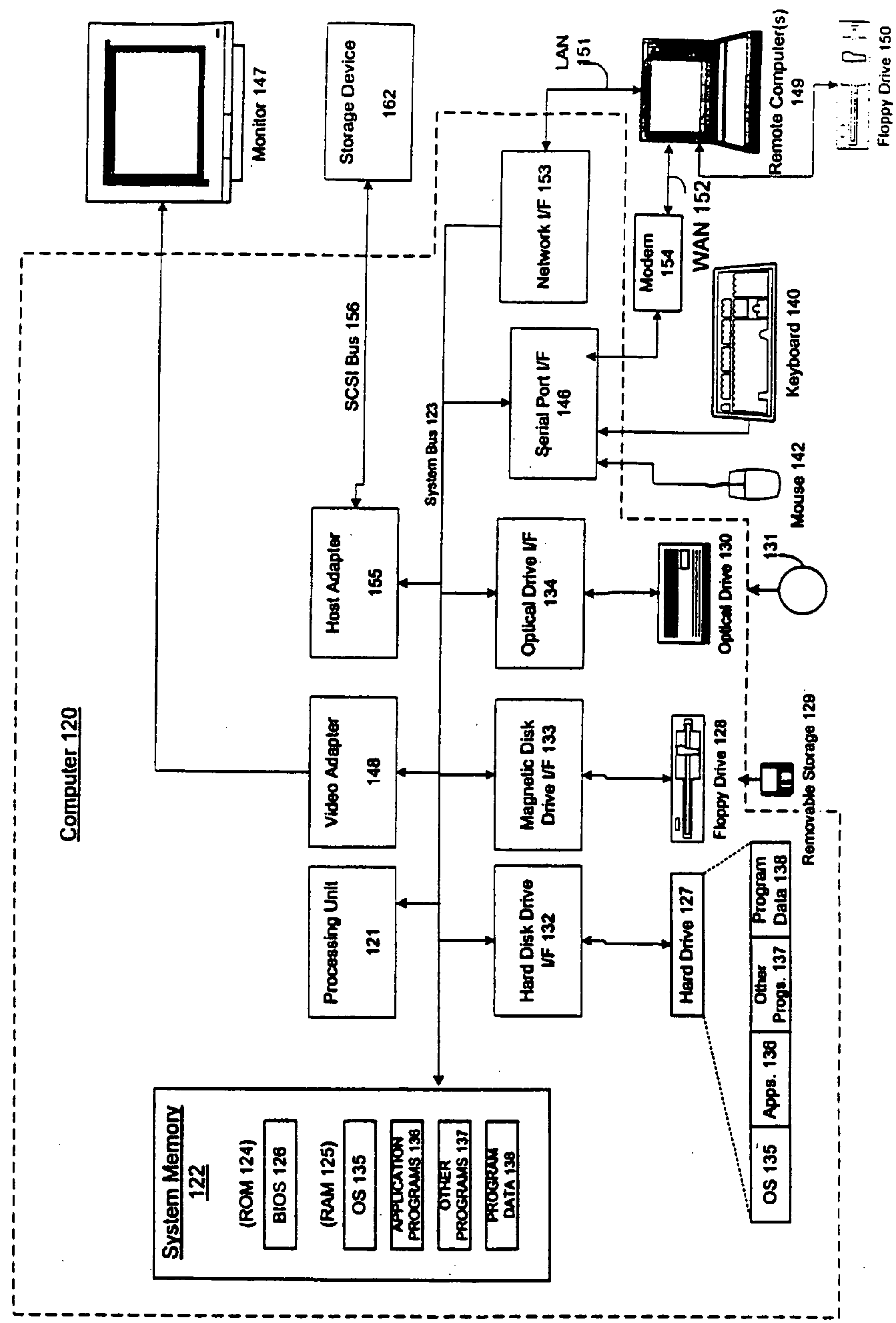

Computer 120
System Memory 122
(ROM 124)
BIOS 126
(RAM 125)
OS 135
APPLICATION PROGRAMS 136
OTHER PROGRAMS 137
PROGRAM DATA 138
Processing Unit 121
Video Adapter 148
Host Adapter 155
System Bus 123
SCSI Bus 156
Storage Device 162
Monitor 147
Hard Disk Drive I/F 132
Magnetic Disk Drive I/F 133
Optical Drive I/F 134
Serial Port I/F 146
Network I/F 153
Hard Drive 127
Floppy Drive 128
Optical Drive 130
Removable Storage 129
131
OS 135
Apps. 136
Other Progs. 137
Program Data 138
Mouse 142
Keyboard 140
Modem 154
WAN 152
LAN 151
Remote Computer(s) 149
Floppy Drive 150

Fig. 12

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 November 2000 (30.11.2000)

PCT

(10) International Publication Number
WO 00/72118 A1

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/US00/10213

(22) International Filing Date: 13 April 2000 (13.04.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/316,717 21 May 1999 (21.05.1999) US

(71) Applicant: COMPAQ COMPUTERS INC. [US/US]; 10435 N. Tautau Avenue, Loc 200-16, Cupertino, CA 95014-3548 (US).

(72) Inventors: GLASSMAN, Steven, C.; 615 Palo Alto Avenue, Mountain View, CA 94041 (US). MANASSE, Mark, S.; 1270 Monterey Boulevard, San Francisco, CA 94127 (US).

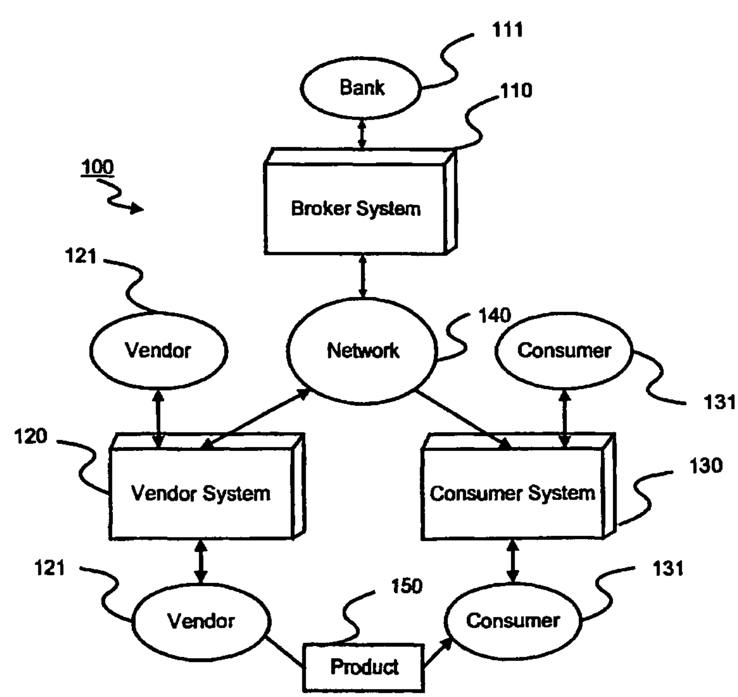
(74) Agents: GRANATELLI, Lawrence; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 et al. (US).

(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, 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, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: 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,

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR ENFORCING LICENSES ON AN OPEN NETWORK



(57) Abstract: An electronic commerce system and method enforces a license agreement for content on an open network (140) by restricting the number of consumers (131) that can concurrently access the content. A consumer (131) initially acquires vendor scrip, either from a broker or the vendor (121) itself. The consumer (131) presents the vendor scrip to the vendor (121) along with a request to access the content. In response, the vendor (121) gathers information about the consumer (131) to determine whether the consumer (131) belongs to the class allowed to access the content. The information may be gathered from the scrip or from other sources. If the consumer (131) belongs to the class, then the vendor (121) determines if a license to access the content is available. Generally, a license is available if the number of other consumers (131) having licenses to access the content is less than the maximum specified in the license agreement. If no licenses are available, the vendor (121) provides the consumer (131) with an estimate of when a license will be available. If a license is available, the vendor (121) directs the consumer (131) to obtain license scrip which allows the consumer (131) to access the content. The license scrip expires after a relatively brief period of time. When the consumer (131) uses the license scrip to access the content, the vendor (121) provides the consumer (131) with new license scrip having a later expiration time.

WO 00/72118 A1

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.

METHOD AND SYSTEM FOR ENFORCING LICENSES ON AN OPEN NETWORK

BACKGROUND

FIELD OF THE INVENTION

This invention relates generally to an electronic commerce system and more particularly to a commerce system supporting restricted use of a resource, and even more particularly to a commerce system supporting N-user license agreements.

BACKGROUND OF THE INVENTION

It is common for a library, corporation, or other organization to purchase content that will be made available to members of the organization. Often, the content is subject to a license restriction limiting distribution of the content. For example, a corporation may license or purchase a magazine and then distribute the magazine to interested employees. Typically, the corporation is restricted by the licensing agreement or copyright law from photocopying the magazine. Accordingly, the corporation must either obtain multiple copies of the magazine or circulate the single copy through the organization.

Similarly, the content licensed or purchased by the organization may be in electronic form. For example, the corporation may license a CD-ROM holding an electronic version of the magazine. While the CD-ROM can be loaded onto a server accessible to employees of the corporation via a computer network, the content may be restricted by an N-user license that forbids the corporation from allowing more than N users to simultaneously access the CD-ROM. To implement the restriction, software executing on the server tracks the number of people currently accessing the CD-ROM and blocks usage that exceeds the scope of the license.

In existing systems, the license control is performed by a combination of a specialized lock server and a client program. The lock server validates users' requests for access to the content and maintains the status of active users. The client program interacts with the lock server to acquire a lock and to provide access to the content.

There are many existing implementations of lock servers. However, they all are subject to one or more of the following undesirable restrictions:

each content source has its own, separate, and proprietary lock server;

the user's system already has the content (protected from direct access) and the client program gets the lock to access the content;

acquiring a lock is a complicated action; and/or

the set of valid users is limited.

For these reasons, existing lock servers are undesirable on an open network.

A lock server providing an N-user license on an open network should also support the following requirements:

an unrestricted set of potential users;

no single administrative domain covers all users;

the users do not need to have a separate user application for each source of content;

access to the content can be easily restricted; and

the content exists on the server and not with the user.

Accordingly, there is a need for a way to provide restricted access to electronic content that works with a wide variety of possible access schemes. Preferably, the solution will allow enforcement of an N-user license for content located on an open network like the Internet.

SUMMARY OF THE INVENTION

The above needs are met by a method and system for electronic commerce that uses special scrip - called "license scrip" - to provide temporary licenses to consumers accessing content. Scrip is primarily used as a form of electronic currency, however it can be more generally considered as a one-time token representing a general value. When scrip is used as an electronic currency, its value is monetary. When scrip is used as a temporary license, its value is the permission to access specific content. This permission may be unlimited or it may be for only a relatively brief period of time, say a few minutes to a few hours.

Accessing content with license scrip is very much like buying regular content with monetary scrip. Instead of having a price specified in monetary terms. Each page of content has a price (which may be zero) given in terms of license scrip. A consumer obtains license scrip from the vendor, preferably exchanging regular vendor scrip for the license scrip.

The vendor uses the license scrip to enforce an N-user license agreement - granting up to N people simultaneous access to the content. The vendor tracks the number and identity of consumers currently having licenses to access the content (i.e., consumers currently possessing valid license scrip).

A consumer initially lacks the license scrip needed to access the content. Upon receiving an access request from the consumer, the vendor determines whether a license is available. If a license is not available, the vendor tells the consumer to try again later and, optionally, provides the consumer with an estimate of when a license will be available.

If a license is available, then the vendor directs the consumer to obtain license scrip. Normally, the consumer obtains license scrip by requesting it from the vendor, but the consumer may get the license by any acceptable means. After receiving a license scrip request. the vendor verifies that the consumer belongs to a class entitled to have a license. For example, if licenses are available to residents of only a certain state, the vendor ensures that the consumer resides in the state before granting the consumer a license.

If a license is available, then the vendor provides the consumer with the license scrip and remembers the granted license. The license scrip is preferably set to expire after a brief time period, but the duration of the license may vary depending upon business or legal concerns. To access content covered by the license, the consumer provides the license scrip when requesting content from the vendor. Each time the consumer accesses the content, the vendor returns replacement license scrip having the same or a later expiration time. Accordingly, the consumer can access the content as long as their license remains valid. When the consumer has not accessed the content for a while, the license scrip expires and the consumer can no longer access the content without obtaining new license scrip.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a top-level block diagram illustrating a computerized system for conducting electronic commerce;

FIGURE 2 is a block diagram illustrating a computer system used in the system of FIG. 1;

FIGURE 3 is a flow diagram illustrating the operations of the system of FIG. 1;

FIGURE 4 is a block diagram illustrating the data fields of a piece of scrip used in the system of FIG. 1;

FIGURE 5 is a diagram illustrating transactions between a consumer and a vendor utilizing license scrip to enforce an N-user license agreement according to the present invention; and

FIGURE 6 is a flow chart illustrating steps for determining whether to grant a license to a consumer.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A preferred embodiment of the present invention restricts access to electronic content through the use of an electronic commerce system. Accordingly, it is useful to describe the electronic commerce system before detailing how the system is utilized according to the present invention.

FIG. 1 shows a computerized system 100 for conducting electronic commerce. The system 100 includes a broker system 110, a vendor system 120, and a consumer system 130 interconnected by a communications network 140.

For clarity, the system 100 depicted in FIG. 1 shows only single broker, vendor, and consumer systems. In actual practice, any number of broker, vendor, and consumer systems can be interconnected by the network 140. The network 140 can be public or private, such as, for example, the Internet, an organization's intranet, a switched telephone system, a satellite linked network, or another form of network. The broker 111 using the broker system 110 can be a bank, a credit provider, an Internet service provider, a telephone company, or any institution the consumer trusts to sell electronic currency called "scrip."

The vendor system 120 is operated by a vendor 121. The vendor 121 provides products and/or content 150 of any type to consumers and, in one embodiment, provides content which is available by subscription. Each subscription page (i.e., page of data that is available for "purchase") has a price of zero but requires a special type of scrip, called "subscription scrip," before it can be accessed. Since the price of a page is zero, the consumer 131 can "purchase" an unlimited number of pages once the consumer 131 has the proper subscription scrip 330. The subscription expires when the subscription scrip 330 expires.

A consumer 131 can use the consumer computer system 130 to electronically acquire the products or content 150 of the vendor 121. As used herein, "consumer" refers to an organization such as a library or corporation, a member of the organization, such as a librarian or an employee, or an individual, such as a person visiting a library or a home computer user. Of course, actions attributed to the organization are usually performed by a member of the organization.

A computer system 200 suitable for use as the broker, vendor, and consumer systems is shown in FIG. 2. The computer system 200 includes a central processing unit (CPU) 210, a memory 220, and an input/output interface 230 connected to each other by a communications bus 240. The CPU 210, at the direction of users 250, e.g. brokers, vendors, and/or consumers, executes software programs, or modules, for manipulating data. The programs and data can be stored in the memory 220 as a database (DB) 221. The DB 221 storing programs and data on the consumer computer system 130 is referred to as a "wallet." In a preferred embodiment of the present invention described herein, many of the operations attributed to the consumer are, in fact, performed automatically by the wallet 221.

The memory 220 can include volatile semiconductor memory as well as persistent storage media, such as disks. The I/O interface 230 is for communicating data with the network 140, the users 250, and other computer system peripheral equipment, such as printers, tapes, etc.

The computer system 200 is scaled in size to function as the broker, vendor, or consumer systems. For example, when scaled as the consumer computer system 130, the computer system 200 can be a small personal computer (PC), fixed or portable. The configurations of the computer system 200 suitable for use by the broker 111 and the vendor 121 may include multiple processors and large database equipped with "fail-safe" features. The fail-safe features ensure that the database 221 is securely maintained for long periods of time.

FIG. 3 shows an operation of the electronic commerce system 100. The consumer 131 uses currency to purchase electronic broker scrip 320 generated by the broker 111. Here, purchasing means that upon a validation of the authenticity of the consumer 131 and the consumer's currency 310, the broker system 110 generates signals, in the form of data records. The signals are communicated, via the network 140, to the consumer system 130 for storage in the wallet 221 of the memory 220 of the consumer system 130.

The scrip is stamped by the generator of the scrip to carry information that is verifiable by the originator, and any other system that has an explicit agreement with the originator. In addition, each scrip is uniquely identifiable and valid at only a single recipient. After a single use, the recipient of the scrip can invalidate it, meaning that the signals of the data record are no longer accepted for processing by the recipient computer system.

In one embodiment, the consumer 131 exchanges the broker scrip 320 with the broker 111 for vendor scrip 330. To complete this transaction, the broker system 110 executes licensed software programs which generate scrip 330 for consumers as needed. Alternatively, the broker 111, in a similar transaction 303, exchanges currency 310 for bulk vendor scrip 330 which is then sold to consumers.

In another embodiment, the consumer 131 exchanges currency with the vendor 121 for regular vendor. In this latter embodiment, there is no need for a broker 111. In addition, the vendor scrip may be free, meaning that the consumer 131 does not need to exchange currency for the scrip.

The consumer 131, in a transaction 304, provides the scrip 330 to the vendor 121. The vendor 121 checks the stamp of the scrip 330 to verify its authenticity, and also checks to make sure the value of the scrip covers the requested content and has not expired. Approval of the transaction results in the delivery of the desired content 150 to the consumer 131. The vendor 121 can also return 304 modified scrip 330 to the consumer 131 as change.

FIG. 4 is a block diagram illustrating the data fields of a single piece of scrip 400. The scrip 400 is logically separated into seven data fields. The Vendor field 410 identifies the vendor for the scrip 400. The Value field 412 gives the value of the scrip 400. The scrip ID field 414 is the unique identifier of the scrip. The Customer ID field 416 is used by the broker 111 and vendor 121 to verify that the consumer has the right to spend the scrip. The Expires field 418 gives the expiration time for the scrip 400. The Props field 420 holds consumer properties, such as the consumer's age, state of residence, employer, etc. Finally, the Stamp field 422 holds a digital stamp and is used to detect tampering with the scrip 400.

The present invention uses "license" scrip, which can be thought of as special purpose scrip having a short period of validity. A consumer with license scrip has a license to view the content covered by the license until the scrip expires.

FIG. 5 is a diagram illustrating transactions between a consumer 510 and a vendor 512 utilizing license scrip to enforce an N-user license agreement according to the present invention. In the transactions of FIG. 5, the vendor 512, for example, can be a library located at a state university. Assume the library purchases a four user license for a CD-ROM and makes the CD-ROM available to other terminals in the library via a local area network and residents of the state via the Internet. To conform with the license, the library must ensure that no more than four consumers are simultaneously accessing the CD-ROM. In this example, the library is the vendor 512 and the people who can access the CD-ROM, either in the library or elsewhere, are the consumers 510.

In another example, a newspaper publisher operates a web site. Assume that a corporation purchases a 20 user license allowing up to 20 people from the corporation to simultaneously access content on the web site. To police its license, the publisher tracks the users of its web site and block users who are not licensed or who have exceeded the scope of the applicable license. Accordingly, the newspaper publisher is the vendor 512 and the corporation and its employees are the consumers 510.

Although neither the illustrated transactions nor the above examples directly utilize a broker, there may be circumstances where it is desirable to use a broker 111 to perform one or more of the transactions described below. Those of ordinary skill in the art will understand that certain transactions attributed to the consumer or the vendor can be performed instead by a broker 111. For example, the library and/or newspaper may issue vendor and license scrip directly or rely on a third-party broker for this task.

Turning to FIG. 5, the consumer 510 initially requests 520 content from the vendor 512 without valid license scrip. In response, the vendor 512 checks to determine whether there is an available license (i.e., whether an additional consumer is allowed to view the content under the license). Preferably, the vendor 512 maintains a data structure associated with the licensed content that can be quickly scanned to determine whether a license is available. In one embodiment, this data structure is a simple N-entry array, with each entry holding fields for the expiration time and Customer ID of the consumer 510 having the license. As licenses are granted, the vendor 512 fills in the array until no more entries are available.

If no licenses are available, then the vendor 512 instructs 522 the consumer 510 to try again later. In one embodiment, the vendor 512 scans the data structure to determine when the first license may become available and provides the consumer 510 with that time as a suggestion of when to try to access the content again. If a license is available, then the vendor 512 instructs the consumer 510 to go and obtain license scrip.

In response, the consumer 510 attempts 524 to obtain license scrip from the vendor 512. The vendor 512 determines whether the consumer 510 is entitled to a license (i.e., entitled to view the content). FIG. 6 is a flow chart 600 illustrating steps for determining whether to grant license scrip to the consumer 510. When the vendor 512 receives the request from the consumer 510, the vender retrieves 610 information about the consumer. The vendor 514 may retrieve this information by asking the consumer 510 to provide it, from the scrip used to request the license scrip, from a "cookie" on the consumer's computer system, or from a table of information shared by the vendor 512 and the consumer 510 or a broker 111. Additionally, the wallet 221 on the consumer's computer system 130 may be configured to automatically provide information about the consumer 510 when requested by a vendor 512. Depending on the needs of the vendor 512 and the license agreement for the content, the information that may be gathered in this manner includes whether the consumer 510 is a member of an organization, the state of residence of the consumer, the consumer's age, or any other information that is relevant to determining whether to provide access to the consumer 510.

The vendor 512 uses this information to determine 612 whether the consumer belongs to a class that has access to the content held by the vendor 512. If the consumer does not belong to a class having access, for example, if the consumer is not a state resident, then the vendor denies 614 access to the consumer 510. Preferably, the vendor 512 directs the consumer 510 to a web page explaining why access was denied.

If the consumer 510 belongs to a class having access, the vendor 512 scans the data structure identifying the current licensees of the content and determines 616 whether an additional license is available. Since there may be a delay between the time the consumer 510 is told to buy license scrip and when the wallet 221 tries to buy the scrip, it is possible that the available license may have been acquired by another consumer during that time. If no licenses are available, then the consumer 510 is told to try again later and optionally given a time when a license may be available.

If a license is available, then the vendor 512 grants 618 the license to the consumer 510. The vendor 512 provides 526 the consumer with license scrip that allows the consumer 510 to access the content. The license scrip preferably has a relatively short validity period, say a few minutes to an hour, and allows the consumer 510 full access to the licensed material for the duration of the scrip. The choice of expiration time for the scrip is a business or legal decision. Since the intention of the license scrip is to hold onto one license slot while the consumer 510 is actively using the content, the duration of the license should cover the time that the consumer 510 is expected to be active. In another embodiment, the duration of the scrip is determined, at least in part, by the type of content accessed by the consumer 510. In addition, the vendor 512 preferably records data about the granted license, including the Customer ID of the consumer 510 and the expiration time of the license in the appropriate data structure.

Each time the consumer 510 wishes to access 528 content held by the vendor 512, the consumer provides the license scrip to the vendor. If the scrip is expired or otherwise invalid. then the consumer's request for access is treated as a request without scrip as illustrated by transaction 520. If the scrip is valid, then the vendor 512 allows the consumer 510 to access the content. In addition, the vendor 512 provides 530 the consumer 510 with replacement license scrip having an updated expiration time. Typically, the updated expiration time is later than the old expiration time, although it can be the same or earlier. In one embodiment, the vendor 512 grants the consumer 510 less additional time each time the vendor issues new license scrip to ensure that the consumer's license eventually expires and other consumers may eventually access the content. The vendor 512 also updates its data structure to reflect the new expiration date of the consumer's license.

Periodically, the vendor 514 preferably scans the data structure to determine whether any licenses have expired. If so, the entry is purged from the data structure, thereby freeing up a license for another consumer 510. Accordingly, the present invention uses license scrip to enforce an N-user license agreement.

It should be understood that FIG. 5 illustrates only one possible set of transactions. FIG. 3. in combination with FIG. 5, provides insight into other possible transactions. For example, a corporation could purchase an N-user license agreement from a broker 111 to access content on a vendor's system 120. The broker 111 can verify that the corporation is entitled to a license and then issue the license scrip from a special scrip series corresponding to the number of users covered by the license. The vendor 121 knows from the scrip series to restrict access from consumers using that license scrip.

Having described a preferred embodiment of the invention, it will now become apparent to those skilled in the art that other embodiments incorporating its concepts may be provided. It is felt therefore, that this invention should not be limited to the disclosed invention, but should be limited only by the spirit and scope of the appended claims.

CLAIMS

We claim:

1. A method of restricting simultaneous access to content, comprising the steps of:

receiving a request to access the content from a consumer;
determining whether the consumer is entitled to access the content; and
responsive to a positive determination, providing the consumer with license scrip allowing access to the content.

2. The method of claim 1, wherein the request to access the content is accompanied by license scrip having an expiration time and wherein the providing step provides the consumer with additional license scrip having an updated expiration time.

3. The method of claim 1, wherein the license scrip has an expiration time and further comprising the steps of:

receiving a second request to access the content from the consumer, the second request including the license scrip; and
responsive to the second request, providing the consumer with replacement license scrip having an updated expiration time.

4. The method of claim 1, wherein the step of determining whether the consumer is entitled to access the content comprises the steps of:

determining whether the consumer belongs to a class having access to the content; and
determining whether a license to access the content is available.

5. The method of claim 4, wherein the step of determining whether the consumer belongs to a class having access to the content comprises the step of:

determining information about the consumer from scrip utilized to request access to the content.

6. The method of claim 4, wherein the step of determining whether a license to access the content is available comprises the steps of:

determining a number of consumers that have licenses to access the content; and

determining a number of allowed licenses;

wherein a license to access the content is available if the number of consumers that have licenses to access the content is less than the number of allowed licenses.

7. The method of claim 4, further comprising the step of:

responsive to a determination that no licenses to access the content are available, providing the consumer with an estimate of when a license will be available.

8. A computer program product having computer-readable instructions embodied thereon for restricting access to content stored on a computer system, the computer-readable instructions comprising instructions for:

receiving a request to access the content stored on the computer system, the request accompanied by scrip;

determining whether the scrip authorizes access to the content;

responsive to a determination that the scrip does not authorize access to the content, determining whether scrip authorizing access to the content is available; and

responsive to a determination that scrip authorizing access to the content is available, providing the scrip.

9. The computer program product of claim 8, further comprising instructions for:

responsive to a determination that the scrip authorizes access to the content, providing replacement scrip having an updated expiration time.

10. The computer program product of claim 8, wherein the instructions for determining whether the scrip authorizes access to the content further comprise computer instructions for:

determining a type of the scrip accompanying the request; and

responsive to a determination that accompanying scrip is license scrip,

determining whether the license scrip has expired, wherein unexpired license scrip authorizes access to the content.

11. The computer program product of claim 8, wherein the instructions for determining whether scrip authorizing access to the content is available comprise instructions for:

determining a maximum number of requesters that can be authorized to access the content;

determining whether a current number of requesters authorized to access the content is less than the maximum number of requesters; and

responsive to a determination that the current number of requesters authorized to access the content is less than the maximum number of requesters, determining that scrip authorizing access to the content is available.

12. The computer program product of claim 8, further comprising instructions for:

responsive to a determination that scrip authorizing access to the content is not available, calculating an estimate of when the scrip authorizing access will be available.

13. A computer system for limiting a number of users that can access content stored on a server associated with the computer system, the computer system comprising:

a module for receiving a request from a user to access the content stored on the server;

a module for determining the number of users currently having rights to access the content; and

a module for providing the user with license scrip if the number of users currently having rights to access the content is less than a number of users allowed to access the content, the license scrip granting the user the right to access the content.

14. The system of claim 13, wherein the module for determining the number of users currently having access rights to content comprises:

a module for scanning a data structure stored in a memory of the computer system, the data structure having one or more entries indicating the number of users having access rights to the content.

15. The computer system of claim 14, wherein the data structure indicates when users' rights to access the content expire, further comprising:

a module for purging the entries of users whose right to access the content has expired.

16. The system of claim 13, wherein only a privileged class can access the content, further comprising:

a module for determining whether the user is a member of the privileged class.

17. The system of claim 13, wherein the license scrip grants the user the right to access the content until an expiration time.

18. The system of claim 17, further comprising:

a module for receiving a second request from the user to access the content stored on the server accompanied by the license scrip; and a module for providing the user with replacement license scrip having a later expiration time.

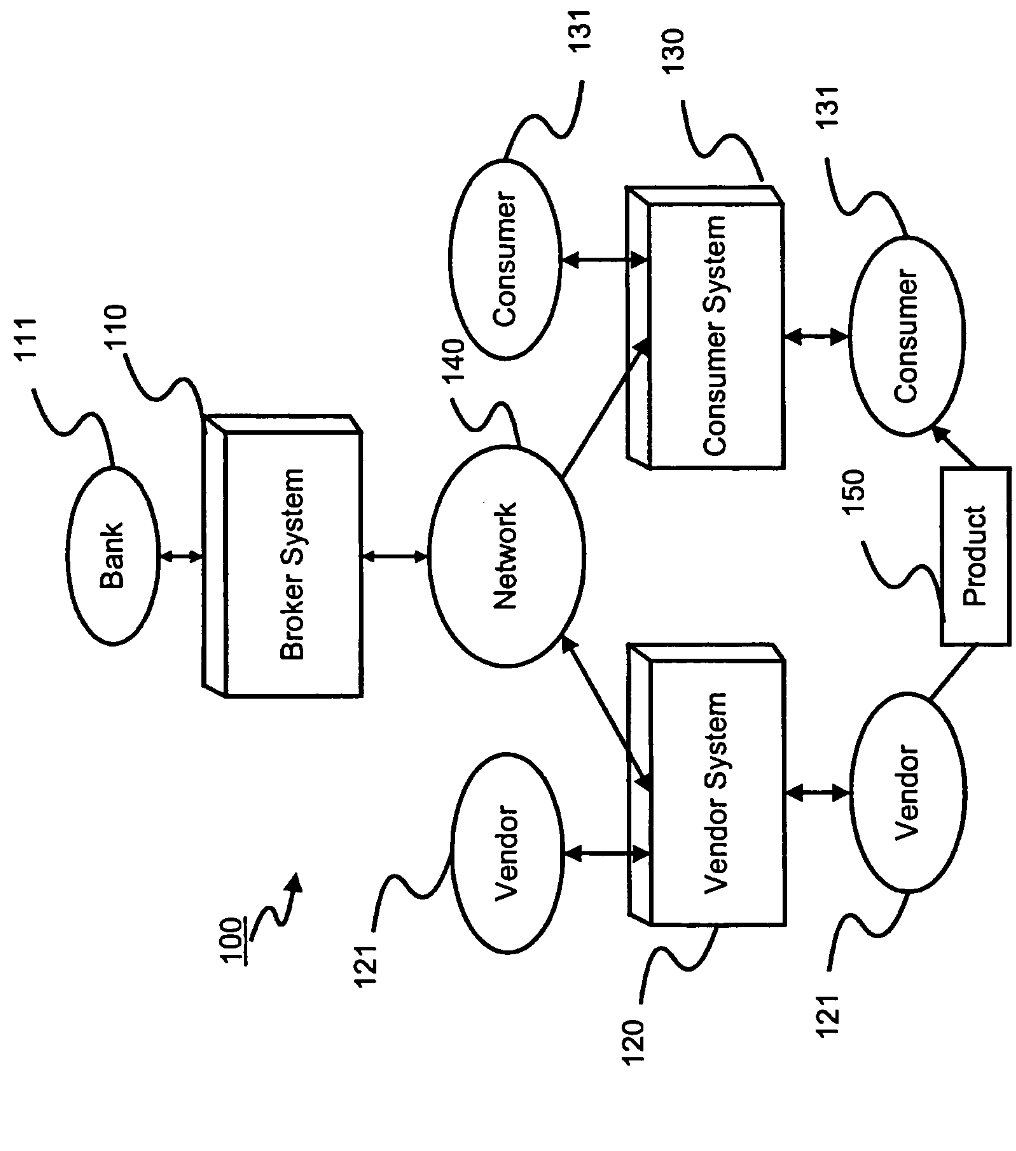

100
111
110
Bank
Broker System
140
Network
131
130
Consumer
Consumer System
131
Consumer
150
Product
121
Vendor
120
Vendor System
121
Vendor

FIG. 1

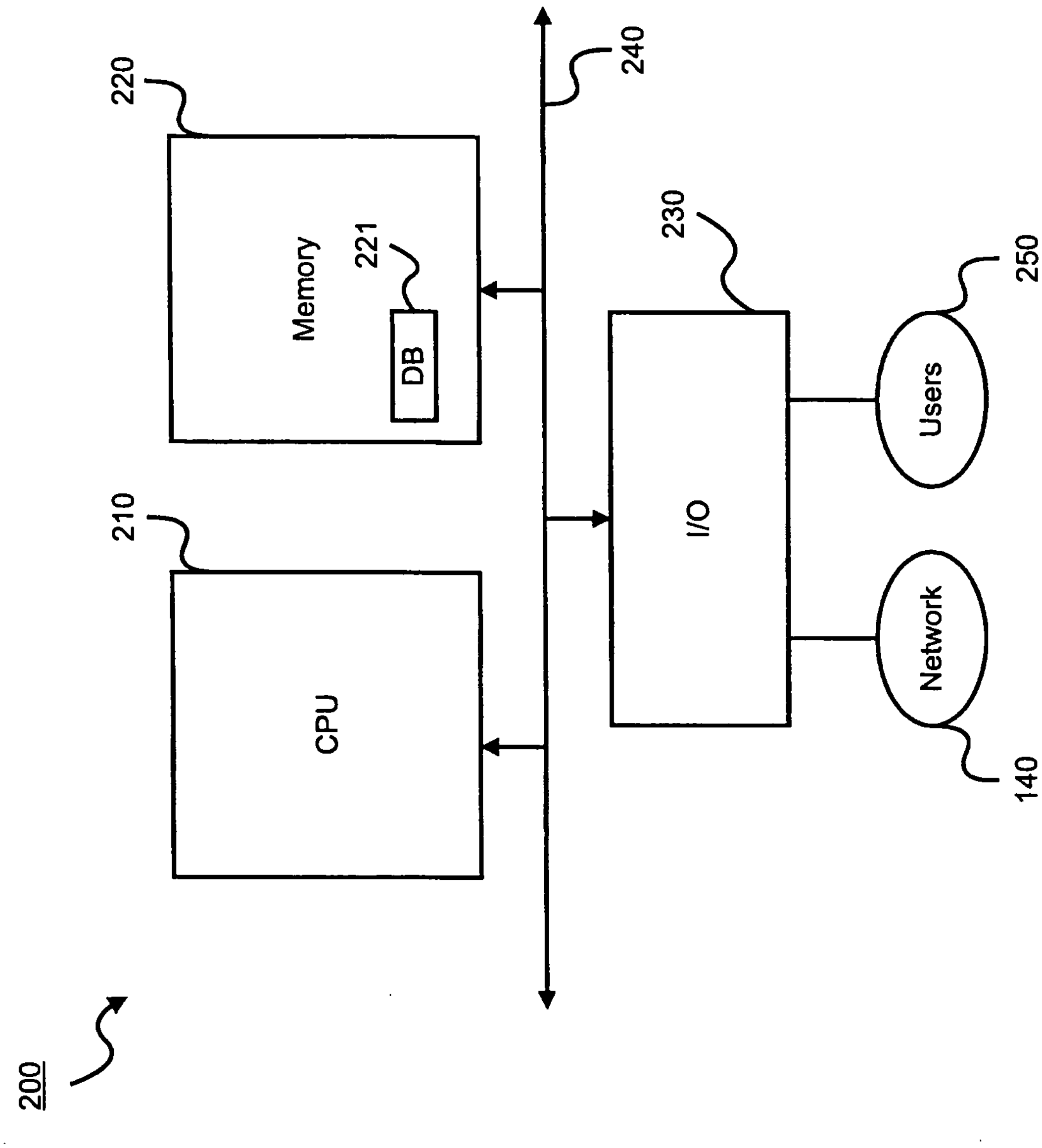


FIG. 2

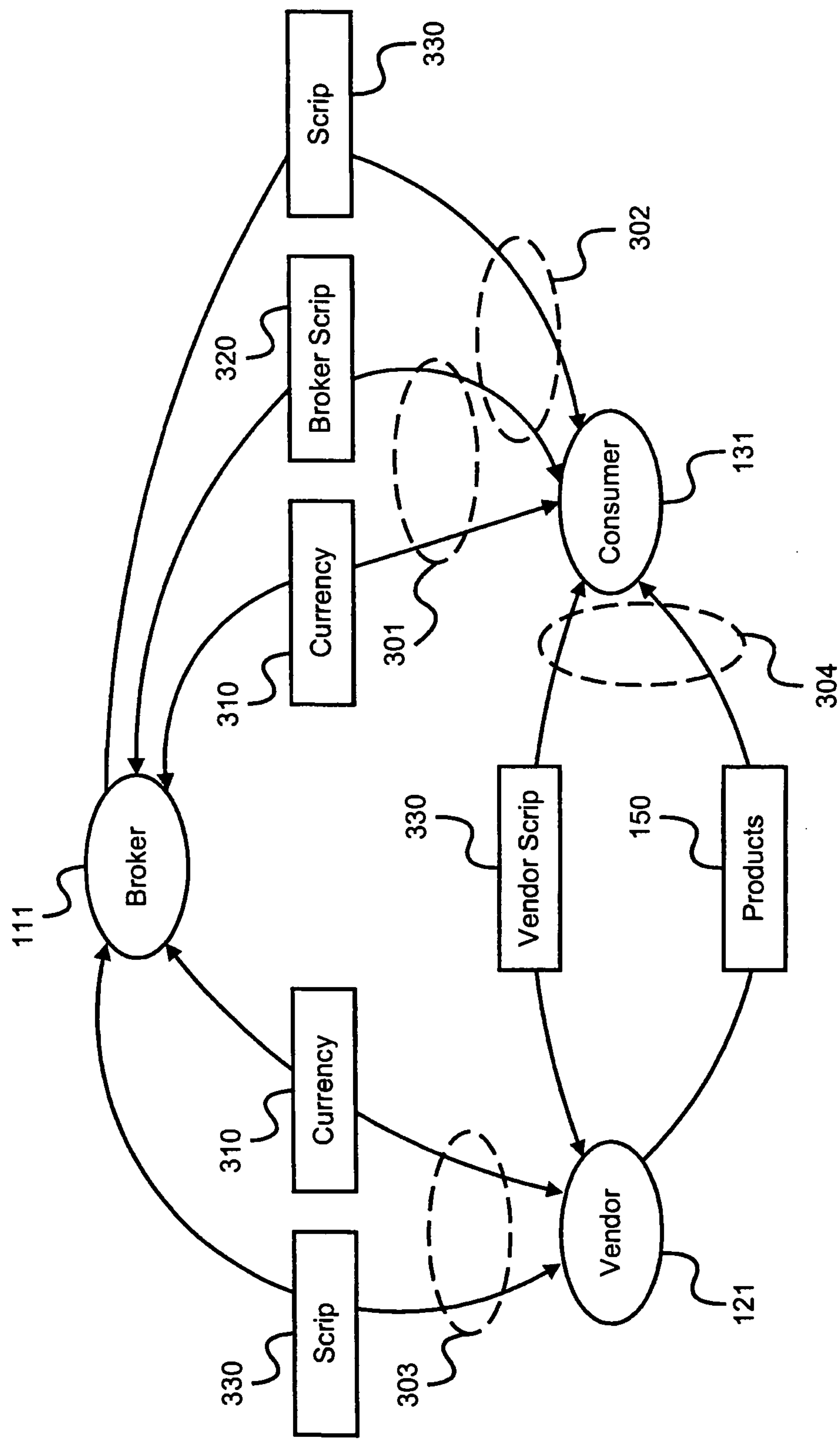

Scrip
330
Broker Scrip
320
Currency
310
Consumer
131
302
301
304
Broker
111
Vendor Scrip
330
Products
150
Currency
310
Scrip
330
303
Vendor
121

FIG. 3

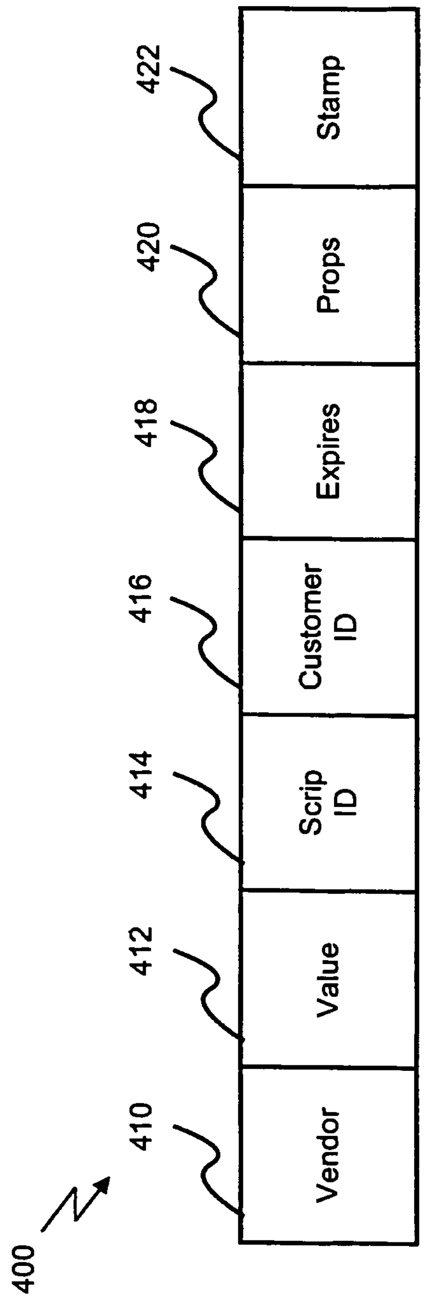

400
410
Vendor
412
Value
414
Scrip ID
416
Customer ID
418
Expires
420
Props
422
Stamp

FIG. 4

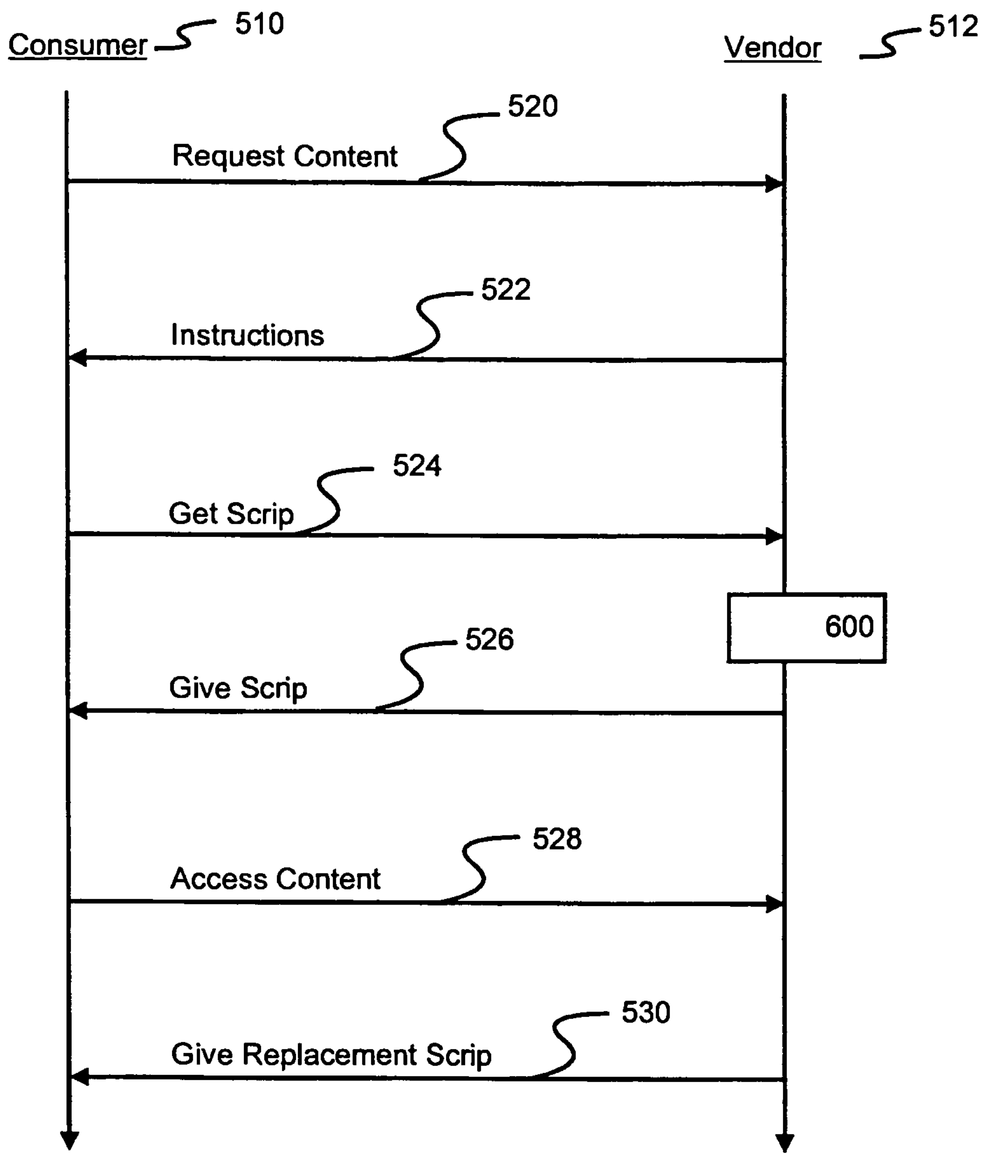
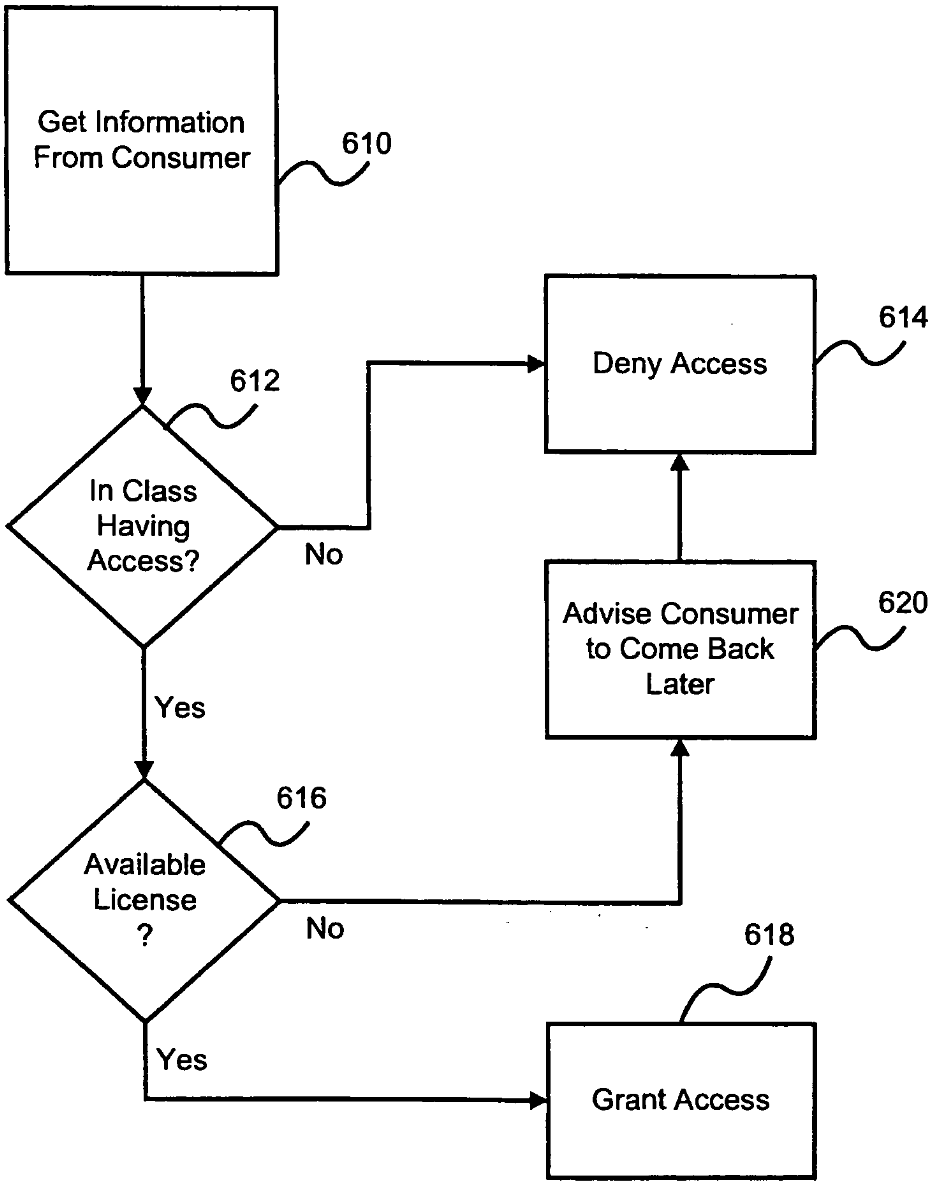

Consumer
510
Vendor
512
520
Request Content
522
Instructions
524
Get Scrip
600
526
Give Scrip
528
Access Content
530
Give Replacement Scrip

FIG. 5


Get Information From Consumer
610
612
In Class Having Access?
No
Yes
Deny Access
614
Advise Consumer to Come Back Later
620
616
Available License ?
No
Yes
618
Grant Access

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/10213

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

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)
IPC 7 G06F G07F

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)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 58306 A (OYLER SCOTT ;GUTHRIE JOHN (US); TECHWAVE INC (US); KRISHNAN GANAPA) 23 December 1998 (1998-12-23)	1,4-6,8, 10,11, 13,14, 16,17
Y		18
A	abstract page 6, line 4 -page 8, line 10 page 10, line 8 -page 16, line 17 page 28, line 7 -page 30, line 23 page 39, line 2 - line 11 figures 1-4	2,3,7,9, 12,15

-/--

[X] Further documents are listed in the continuation of box C.

[X] 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	Date of mailing of the international search report
2 August 2000	09/08/2000

Name and mailing address of the ISA	Authorized officer
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	Jacobs, P

2

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Intern nal Application No
PCT/US 00/10213

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y A	WO 97 03423 A (DIGITAL EQUIPMENT CORP) 30 January 1997 (1997-01-30) page 4, line 15 -page 11, line 33 figures 1-5	1-4,8-10 18 5,13,16, 17
X A	US 5 905 860 A (BRINGHURST ADAM L ET AL) 18 May 1999 (1999-05-18) abstract column 2, line 40 -column 16, line 3	1,4-6, 13,14, 16,17 2,3, 7-12,15, 18
X A	GB 2 316 503 A (ICL PERSONAL SYSTEMS OY) 25 February 1998 (1998-02-25) abstract page 6 -page 21 figures 1-4 claim 1	1,4-6, 13,14, 16,17 2,3, 7-12,15, 18

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 00/10213

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9858306 A	23-12-1998	US 6073124 A	06-06-2000
		AU 8150598 A	04-01-1999
WO 9703423 A	30-01-1997	US 5802497 A	01-09-1998
		BR 9606450 A	30-09-1997
		EP 0796480 A	24-09-1997
		IL 117195 A	20-06-1999
		JP 2984731 B	29-11-1999
		JP 9510814 T	28-10-1997
US 5905860 A	18-05-1999	US 5758069 A	26-05-1998
GB 2316503 A	25-02-1998	NONE	

Form PCT/ISA/210 (patent family annex) (July 1992)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 December 2000 (07.12.2000)

PCT

(10) International Publication Number
WO 00/73922 A2

(51) International Patent Classification[7]: G06F 17/00

(21) International Application Number: PCT/US00/11078

(22) International Filing Date: 25 April 2000 (25.04.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/323,635 1 June 1999 (01.06.1999) US

(71) Applicant: ENTERA, INC. [US/US]; 40971 Encyclopedia Circle, Fremont, CA 94538 (US).

(72) Inventor: SCHARBER, John, M.; 1616 Placer Circle, Livermore, CA 94550 (US).

(74) Agents: FAHMI, Tarek, N. et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, 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, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): 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).

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.

WO 00/73922 A2

(54) **Title:** CONTENT DELIVERY SYSTEM

(57) **Abstract:** Disclosed is a network content delivery system configured to: select a first content routing technique for processing a first set of network content; and select a second content routing technique for processing a second set of network content, wherein the first and second content routing techniques are selected based on one or more content routing variables. Also disclosed is a content delivery system comprising: a network node for storing network content; a first transmission medium communicatively coupled to the network node for transmitting a first set of network content to the network node; and a second transmission medium communicatively coupled to the network node for transmitting a second set of network content to the network node, wherein the first and second sets of network content are selected based on one or more routing variables.

CONTENT DELIVERY SYSTEM

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to the transmission and storage of digital information across a network. More particularly, the invention relates to an improved system and method for caching and/or delivering various types of digital content using a plurality of network protocols.

Description of the Related Art

The World Wide Web (hereinafter "the Web") is a network paradigm which links documents known as "Web pages" locally or remotely across multiple network nodes (i.e., Web servers). A single Web page may have links (a.k.a., "hyperlinks") which point to numerous other Web pages. When a user points and clicks on a link using a cursor control device such as a mouse, the user can jump from the initial page to another page, regardless of where the Web pages are actually located. For example, the initial Web page might be stored on a Web server in New York and the second page (accessed via the hyperlink in the first page) might be stored on a Web server in California.

The underlying principles of the Web were developed 1989 at the European Center for Nuclear Research (CERN) in Geneva. By 1994 there were approximately 500 Web servers on the Internet. Today there are more than a million, with new sites starting up at an extraordinary rate. In sum, the Web has become the center of Internet activity and is the primary reason for the explosive growth of the Internet over the past several years.

In addition to providing a simple point-and-click interface to vast amounts of information on the Internet, the Web is quickly turning into a content delivery system. Well known Internet browsers such as Netscape Navigator™ and Microsoft Internet Explorer™ frequently provide plug-in software which allow additional features to be incorporated into the browser program. These include, for example, support for audio and video streaming, telephony, and videoconferencing.

The unparalleled increase in Web usage combined with the incorporation of high bandwidth applications (i.e., audio and video) into browser programs has created serious performance/bandwidth problems for most Internet Service Providers (hereinafter "ISPs"). Moreover, the network traffic resulting from non-Web-based Internet services such as Internet News (commonly known as "Usenet" News) has increased on the same scale as the increase in Web traffic, thereby further adding to the bandwidth problems experienced by most ISPs.

These issues will be described in more detail with respect to **Figure 1** which illustrates an ISP 100 with a link 160 to a larger network 150 (e.g., the Internet) through which a plurality of clients 130, 120 can access a plurality of Web servers 140-144 and/or News servers 146-148. Maintaining a link 160 to the Internet 150 with enough bandwidth to handle the continually increasing traffic requirements of its clients 120, 130 represents a significant cost for ISP. At the same time, ISP 110 must absorb this cost in order to provide an adequate user experience for its clients 120, 130.

One system which is currently implemented to reduce network traffic across link 160 is a proxy server 210 with a Web cache 220, illustrated in **Figure 2.** When client 120 initially clicks on a hyperlink and requests a Web page (shown as address "www.isp.com/page.html") stored on Web server 144, client 120 will use proxy server 210 as a "proxy agent." This means that proxy server 210 will make the request for the Web page on behalf of client 120 as shown. Once the page has been retrieved and forwarded to client 120, proxy server will store a copy of the Web page locally in Web cache 220. Thus, when client 120 or another client – e.g., client 130 – makes a subsequent request for the same Web page, proxy server 210 will immediately transfer the Web page from its Web cache 210 to client 130. As a result, the speed with which client 130 receives the requested page is substantially increased, and at the same time, no additional bandwidth is consumed across Internet link 160.

While the foregoing proxy server configuration alleviates some of the network traffic across Internet link 160, several problems remain. One problem is that prior Web cache configurations do not have sufficient intelligence to deal with certain types of Web pages (or other Web-based information). For example, numerous Web pages and associated content can only be viewed by a client who pays a subscriber fee. As such, only those clients which provide proper authentication should be permitted to download the information. Today, proxy servers such as proxy server 210 will simply not cache a Web document which requires authentication.

In addition, Web caches do not address the increasing bandwidth problem associated with non-Web based Internet information. In particular, little has been done to alleviate the increasing bandwidth problems created by Usenet news streams. In fact, ISPs today must set aside a substantial amount of bandwidth to provide a continual Usenet news feed to its clients. Moreover, no mechanism is currently available for caching other data transmissions such as the streaming of digital audio and video. The term "streaming" implies a one-way transmission from a server to a client which provides for uninterrupted sound or video. When receiving a streaming transmission, the client will buffer a few seconds of audio or video information before it starts sending the information to a pair of speakers and/or a monitor, thus compensating for momentary delays in packet delivery across the network.

Accordingly, what is needed is a content delivery system which will reduce the bandwidth requirements for ISP 110 while still providing clients 120, 130 with an adequate user experience. What is also needed is a system which will work seamlessly with different types of Web-based and non-Web-based information and which can be implemented on currently available hardware and software platforms. What is also needed is an intelligent content delivery system which is capable of caching all types of Web-based information, including information which requires the authentication of a client before it can be accessed. What is also needed is a content delivery system which is easily adaptable so that it can be easily reconfigured to handle the caching of new Internet information and protocols. Finally, what is needed is a data replication system which runs on a distributed database engine, thereby incorporating well known distributed database procedures for maintaining cache coherency.

SUMMARY OF THE INVENTION

Disclosed is a network content delivery system configured to: select a first content routing technique for processing a first set of network content; and select a second content routing technique for processing a second set of network content, wherein the first and second content routing techniques are selected based on one or more content routing variables.

Also disclosed is a content delivery system comprising: a network node for storing network content; a first transmission medium communicatively coupled to the network node for transmitting a first set of network content to the network node; and a second transmission medium communicatively coupled to the network node for transmitting a second set of network content to the network node, wherein the first and second sets of network content are selected based on one or more routing variables.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the present invention can be obtained from the following detailed description in conjunction with the following drawings, in which:

FIG. 1 illustrates generally a network over which an ISP and a plurality of servers communicate.

FIG. 2 illustrates an ISP implementing a proxy server Web cache.

FIG. 3 illustrates one embodiment of the underlying architecture of an Internet content delivery system node.

FIG. 4 illustrates a plurality of Internet content delivery system nodes communicating across a network.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

One embodiment of the present system is a computer comprising a processor and a memory with which software implementing the functionality of the internet content delivery system described herein is executed. Such a computer system stores and communicates (internally or with other computer systems over a network) code and data using machine readable media, such as magnetic disks, random access memory, read only memory, carrier waves, signals, etc. In addition, while one embodiment is described in which the parts of the present invention are implemented in software, alternative embodiments can implement one or more of these parts using any combination of software, firmware and/or hardware.

The underlying architecture of one embodiment of the present internet content delivery system (hereinafter "ICDS") is illustrated in **Figure 3**. A single ICDS node 300 is shown including a cache 330, an ICDS application programming interface (hereinafter "API") 360 which includes a distributed database engine 361, and a plurality of software modules 310-326 which interface with the ICDS API 360. ICDS node 300 may communicate over a network 340 (e.g., the Internet) over communication link 370 and may also interface with a plurality of clients 350-351 and/or other ICDS nodes (e.g., through link 380).

As is known in the art, an API such as ICDS API 360 is comprised of a plurality of subroutines which can be invoked by application software (i.e., software written to operate in conjunction with the particular API). Thus, in **Figure 3** each of the plurality of software modules 310-326 may be uniquely tailored to meet the specific needs of a particular ISP. The modules interface with API 360 by making calls to the API's set of predefined subroutines. Another significant feature of ICDS API 360 is that it is platform-independent. Accordingly, it can be implemented on numerous hardware platforms including those that are Intel-based, Macintosh-based and Sun Microsystems-based.

In one embodiment, a portion of API 360's subroutines and a set of prefabricated modules can be marketed together as a Software Development Kit (hereinafter "SDK"). This will allow ISPs, corporations and/or end-users to customize the type of internet content delivery/caching which they require. In addition, because modules 310-326 may be dynamically linked, they may be loaded and unloaded without having to reboot the hardware platform on which cache 330 is executed.

I. Distributed Content Processing

As illustrated, ICDS node 300 includes a plurality of network protocol modules 310-319 which interface with API 360. These modules provide caching support on ICDS node 300 for numerous different Internet protocols including, but not limited to, Web protocols such as the Hypertext Transfer Protocol (hereinafter "HTTP") 310, Usenet news protocols such as the Network News Transport Protocol (hereinafter "NNRP") 312, directory access protocols such as the Lightweight Directory Access Protocol (hereinafter "LDAP") 314, data streaming protocols such as the Real Time Streaming Protocol (hereinafter "RTSP") 316, and protocols used to perform Wide Area Load Balancing (hereinafter "WALB") 318. Because the underlying architecture of the present ICDS system includes an open API, new protocol modules (e.g., module 319) can be seamlessly added to the system as needed.

One embodiment of the ICDS system includes a plurality of standardized service definitions through which individual service modules 320-326 may be configured to interface with the ICDS API 360. These service modules provide the underlying functionality of ICDS node 300 and may include a data services module 320, an access services module 322, a transaction services module 323, a commercial services module 324, a directory services module 325, and a resource services module 326. The functionality of each of these modules will be described in more detail below.

In one embodiment of the ICDS system, the ICDS API includes a distributed relational database engine 361. As a result, a plurality of ICDS nodes 410-440 can be distributed across ISP 400's internal network and still maintain a coherent, up-to-date storage of Internet content. For example, if a particular data object is updated at two nodes simultaneously, the underlying distributed database system may be configured to resolve any conflicts between the two modifications using a predefined set of distributed database algorithms. Accordingly, the present system provides built in caching support for dynamically changing Internet content (e.g., Web pages which are modified on a regular basis). Such a result was not attainable with the same level of efficiency in prior art caching systems such as proxy server 210 of **Figure 2** (which are executed on, e.g., standard flat file systems such as UNIX or NFS file servers).

Data Services

Data services modules such as module 320 running on each ICDS node 410-440 provide support for data replication and distribution across ISP 400's internal network 480. This includes caching support for any data protocol included in the set of protocol modules 310-319 shown in **Figure 3** as well as for any future protocol which may be added as a module to the ICDS API 360. Because the ICDS API 360 provides a set of standardized service definitions for data services module 320, an ISP using a plurality of ICDS nodes 300 as illustrated in **Figure 4** can replicate data across its network without an extensive knowledge of distributed database technology. In other words, the ISP can configure its plurality of nodes by invoking the standardized service definitions associated with data services module 320 and leave the distributed database functionality to the distributed database engine 361.

Generally, three different types of data replication may be implemented by the present system: dynamic replication, database replication (or "actual" replication), and index replication. Using dynamic replication, if client 472 requests content from internal ICDS server 460 or from a server across network 490, the content will be delivered to client 472 and replicated in ICDS node 430. If client 473 (or any other client) subsequently requests the same content, it will be transmitted directly from ICDS node 430 rather than from its original source (i.e., a second request to server 460 or a server across network 490 will not be required). Accordingly, bandwidth across ISP 400's internal network and across Internet link 405 is conserved.

The dynamic replication mechanism just described works well for replicating static content but not for replicating dynamically changing content. For example, if the replicated content is a magazine article then caching a copy locally works well because it is static information – i.e., there is no chance that the local copy will become stale (out of date). However, if the replicated content is a Web page which contains continually changing information such as a page containing stock market quotes, then dynamic replication may not be appropriate. No built in mechanism is available for proxy cache server 210 to store an up-to-date copy of the information locally.

The present ICDS system, however, may use database replication to maintain up-to-date content at each ICDS node 410-440. Because the present system includes a distributed database engine 361, when a particular piece of content is changed at one node (e.g., ICDS server 460) a store procedure may be defined to update all copies of the information across the network. This may be in the form of a relational database query. Thus, the present system may be configured to use dynamic replication for static content but to use database replication for time-sensitive, dynamically changing content.

The third type of database replication is known as index replication. Using index replication a master index of content is replicated at one or more ICDS nodes 410-440 across the network 480. Once again, this implementation is simplified by the fact that the underlying ICDS node engine is a distributed database engine. Certain types of information distributions are particularly suitable for using index replication. For example, news overview information (i.e., the list of news articles in a particular newsgroup) is particularly suited to index replication. Instead of replicating each individual article, only the news overview information needs to be replicated at various nodes 410-440 across the network 480. When a client 473 wants to view a particular article, only then will the article be retrieved and cached locally (e.g., on ICDS node 430).

ICDS node 430 is capable of caching and delivering various types of Internet data using any of the foregoing replication techniques. While prior art proxy servers such as proxy server 210 may only be used for caching Web pages, ICDS node 430 is capable of caching various other types of internet content (e.g., news content) as a result of the protocol modules 310-319 interfacing with ICDS API 360. Moreover, as stated above, ICDS node 430 (in conjunction with nodes 410, 420 and 440) may be configured to cache dynamic as well as static Web-based content using various distributed database algorithms.

One specific example of a data service provided by one embodiment of the present system is Wide Area Load Balancing (hereinafter "WALB") using layer 7 switching as described in the co-pending U.S. Patent Application entitled "WIDE AREA LOAD BALANCING" (Serial No. _______), which is assigned to the assignee of the present application and which is incorporated herein by reference. The present system may also perform dynamic protocol selection, dynamic query resolution, and/or heuristic adaptation for replicating content across a network as set forth in the co-pending U.S. Patent Application entitled Dynamic Protocol Selection and "QUERY RESOLUTION FOR CACHE SERVERS" (Serial No. ____), which is assigned to the assignee of the present application and which is incorporated herein by reference. Finally, the present system also may include network news (e.g., Usenet news) services set forth in the co-pending U.S. Patent Applications entitled "HYBRID NEWS SERVER" (Serial No. ___), and "SELF-MODERATED VIRTUAL COMMUNITIES" (Serial No. ___), each assigned to the assignee of the present application and each incorporated herein by reference.

Access Services

As stated above, prior art proxy server cache systems such as proxy server 210 are only capable of caching static, publicly available Web pages. A substantial amount of Web-based and non-Web-based content, however, requires some level of authentication before a user will be permitted to download it. Thus, client 472 (in Figure 4) may pay a service fee to obtain access to content on a particular web site (e.g., from server 460 or from another server over network 490). As a result, when he attempts to access content on the site he will initially be prompted to enter a user name and password. Once the user transmits this information to the Web server, he will then be permitted to download Web server content as per his service agreement.

A problem that arises, however, is that prior art cache systems such as proxy server 210 are not permitted to cache the requested content. This is because proxy server 210 has no way of authenticating subsequent users who may attempt to download the content. Thus, documents which require authentication are simply uncacheable using current network cache systems.

The present ICDS system, however, includes user authentication support embedded in access services module 322. Thus, when client 473, for example, attempts to access a Web page or other information which requires authentication, ICDS node will determine whether the requested content is stored locally. If it is, then ICDS node 430 may communicate with the authentication server (e.g., server 460 or any server that is capable of authenticating client 473's request) to determine whether client 473 should be granted access to the content. This may be accomplished using standardized authentication service definitions embedded in access services module 322. Using these definitions, ICDS node 430 will not only know what authentication server to use, it will also know what authentication *protocol* to use when it communicates to the authentication server. As a result of providing local access services module 322 for authentication, network information which requires authentication can now be cached locally in ICDS node 430, thereby conserving additional bandwidth across network link 405 and/or ISP network 480.

One particular embodiment of the present system replicates Remote Authentication Dial In User Service (hereinafter "RADIUS") information across network 480. RADIUS is an application-level protocol used by numerous ISP's to provide user authentication and profile services. This is achieved by setting up a central RADIUS server with a database of users, which provides both authentication services (i.e., verification of user name and password) and profile services detailing the type of service provided to the user (for example, SLIP, PPP, telnet, rlogin).

Users connect to one or more Network Access Servers (hereinafter "NASs") which operate as a RADIUS clients and communicate with the central RADIUS server. The NAS client passes the necessary user information to the central RADIUS server, and then acts on the response which is returned. RADIUS servers receive user connection requests, authenticate users, and then return all configuration information necessary for the client to deliver service to the user.

One problem associated with the RADIUS protocol is that it does not provide any built in facilities for replication of RADIUS information. Accordingly, on large ISP's such as America Online ("AOL"), which may have tens of millions of users, RADIUS servers are hard hit, potentially handling thousands of logon requests a minute. This may create severe performance/bandwidth problems during high traffic periods. In response, some ISP's have taken a brute-force approach to distributing RADIUS information by simply copying the information to additional servers across the network without any built in mechanism to keep the RADIUS data coherent and up-to-date.

One embodiment of the present ICDS system provides an efficient, dynamic mechanism for distributing RADIUS information. Specifically, a RADIUS module is configured to interface with ICDS API 360 in this embodiment (similar to the way in which protocol modules 310-319 interface with the ICDS API 360). RADIUS information can them be seamlessly distributed across the system using distributed database engine 361. For example, the RADIUS module in conjunction with access services module 322 on ICDS node 430 may maintain radius information for local users. [Exactly how will this work? I assume that access services module will be used but there will be a separate RADIUS protocol module to support the protocol??] Thus, when client 472 first logs in to the system, ICDS node 430 may communicate with a second ICDS node (e.g., central ICDS server 460) which contains the necessary RADIUS authentication and user profile information. Client 472 will input a user name and password and will then be permitted access to the network as per his service agreement with ISP 400.

Unlike previous RADIUS systems, however, ICDS node 430 in the present embodiment may locally cache client 472's RADIUS information so that the next time client 472 attempts to login to the network, the information will be readily available (i.e., no access to a second ICDS node will be necessary). ICDS node 430 may be configured to save client 472's RADIUS information locally for a predetermined period of time. For example, the information may be deleted if client 472 has not logged in to local ICDS node 430 for over a month.

Thus, if client 472 represents a user who frequently travels across the country and logs in to ISP 400's network 480 from various different ICDS nodes, the present system provides a quick, effective mechanism for dynamically replicating client 472's user information into those geographical locations from which he most commonly accesses ISP 400. This reduces the load which would otherwise be borne by a central RADIUS server and also improves client 472's user experience significantly (i.e., by providing him with a quick login).

Database replication can also be used to update RADIUS information distributed across multiple ICDS nodes 410-440. This may be done using known store procedures defined in relational database 361. For example, if client 472 cancels his service agreement with ISP 400, he should not be able to continually log in to local ICDS node 430 using the RADIUS information which has been cached locally. Thus, under the present ICDS system, ISP 400 may simply issue a relational database query such as [let's add another update query here using database terminology as an example] to immediately update ICDS node 430's radius information.

One of ordinary skill in the art will readily recognize from the preceding discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention. Throughout the foregoing description, specific embodiments of the ICDS system were described using the RADIUS protocol in order to provide a thorough understanding of the operation of the ICDS system. It will be appreciated by one having ordinary skill in the art, however, that the present invention may be practiced without such specific details. For example, the ICDS system may also distribute authentication and user profile information in the form of the Lightweight Directory Access Protocol ("LDAP"). In other instances, well known software and hardware configurations/techniques have not been described in detail in order to avoid obscuring the subject matter of the present invention.

Access services module 322 may also provide local encryption/decryption and watermarking of internet content. Audio or video content delivery systems, for example, commonly use encryption of content to protect the rights of the underlying copyright holder. When a user requests a particular piece of content some delivery systems encrypt the content using a unique client encryption key. Only a client who possesses the encryption key (presumably the client who paid for the content) will be permitted to play the content back. Other systems provide for the "watermarking" of content (rather than encrypting) so that the rightful owner of the content may be identified. This simply entails embedding a unique "tag" which identifies the source of the content and/or the owner of the content (i.e., the one who paid for it).

Prior art caching systems such as proxy server 210 are not capable of dealing with encrypted or watermarked content because the encryption/watermarking functionality was not provided locally (i.e., proxy server 210 was not "smart" enough). In one embodiment of the present ICDS system, however, access services module 322 of ICDS nodes 410-440 includes a local encryption module and/or a local watermarking module. For example, if client 473 requests specific content such as a copyrighted music content stored on a music server (e.g., ICDS server 460), the initial request for the content will be made from ICDS node 430 on behalf of client 473. ICDS node 430 will retrieve the requested content and cache it locally. If the requested content requires encryption, ICDS node 430 will use its local encryption module to encrypt the requested content using a unique user encryption key for client 473.

If a second client – e.g., client 472 – requests the same content, the copy stored locally on ICDS module 430 can be used. ICDS module 430 will simply encrypt the content using a *different* encryption key for user 472. Thus, frequently requested multimedia content (which, as is known in the art, can occupy a substantial amount of storage space) may be cached locally at ICDS node 430 notwithstanding the fact that the content requires both user authentication and encryption.

The same functionality may be provided for watermarking of content. ICDS node will use a watermarking module (which may comprise a component of access services module 322) to individually watermark multimedia information requested by individual clients, thereby protecting the rights of the copyright holder of the underlying multimedia content. This information can then be regularly communicated back to a central database repository.

As is known in the art, multimedia files can be extremely large and, accordingly, take substantially more time to communicate across a network than do, for example, generic Web pages. As such, the ability to locally cache multimedia files significantly reduces traffic across network 480, and also significantly improves the user experience for local users when downloading multimedia information.

Transaction Services

In addition to replicating data services and access services information across a network, the present ICDS system also provides for the replication of transaction services. Transaction services includes maintaining information on client payments for use of ISP 400's services as well as information relating to the client's online access profile (i.e., recording of the times when the user is online).

When a client logs in to an ISP today, the client's online information is maintained on a single central server. The central server maintains records of when and for how long the client logged in to the network and may also include information about what the client did while he was online. As was the case with maintaining a central RADIUS server, maintaining a central transaction server for all users of a large ISP is inefficient and cumbersome. The present system solves the performance and bandwidth problems associated with such a configuration by storing transaction information locally via transaction module 323 and algorithms build around distributed database engine 361.

Thus, if client 472 only logs on to ISP 400's network 480 via ICDS node 430, all of his transaction and billing information will be stored locally. The information may then be communicated across network 480 to a central billing server at predetermined periods of time (e.g., once a month). [We didn't go into great detail on transaction services and the rest; please add information as you feel appropriate]

Commercial Services

Commercial services module 324 provides a significantly improved local caching capability for add rotation and accounting. An add rotation system operating in conjunction with a typical proxy cache server will now be described with respect to **Figure 2**. When client 120 downloads a web page from Web server 142 the Web page may contain an ad tag or an add tag may automatically be inserted. The add tag will identify add server 170 and will indicate that an add should be inserted into the requested Web page from add server 170. Add server will then identify a specific add to insert into the downloaded Web page from add content server. The Web page plus the inserted add will then be forwarded to proxy server 210 and on to client 120.

Add server 170 will keep an accounting of how many different users have downloaded Web pages with adds inserted as described above. However, one problem with accounting on this system is that proxy server 210 requests Web pages *on behalf of* its clients. Accordingly, once the requested Web pages has been cached locally in Web cache 220, add server will only receive requests from proxy server 210 for any subsequent requests for the Web page. This will result in an inaccurate accounting of how many unique clients actually requested the Web page (and how many adds were viewed by unique users).

Because one embodiment of the present system provides built in caching support for ad rotation services, an accurate accounting of the number of hits that a particular ad receives may be maintained. More specifically, one embodiment of the present ICDS system solves this problem by providing a commercial services module that monitors and records how often individual clients request Web pages containing particular adds from add content server 171. This information than then be communicated to a central server (e.g., ICDS server 460) at predetermined intervals for generating add rotation usage reports.

Directory Services

Directory services provide the ability to cache locally a directory of information across network 480 or 490. That is, the question here is not whether the particular information is available but where exactly over networks 480 or 490 it is located. It should be noted that there may be some overlap between the directory services concept and the index replication concept described above with respect to data services. [I'm still not 100% sure what this is – please elaborate with an example]

II. <u>Content Routing</u>

The term "content routing" refers to the ability to select among various techniques/protocols for maintaining a coherent set of content across a network. The selection of a particular technique may be based on several routing variables including, but not limited to, the type of content involved (i.e., FTP, HTTP . . . etc), the size of the content involved (i.e., small files such as HTTP vs. large files such as audio/video streaming), the location of the content on network 480 and/or network 490, the importance of a particular piece of content (i.e., how important it is that the content be kept up-to-date across the entire network), the particular user requesting the content and the terms of his subscription agreement (i.e., some users may be willing to pay more to be insured that they receive only the most up-to-date content without having to wait), the frequency with which the content is accessed (e.g., 5%-10% of content on the Internet represents 90% of all the *requested* content), and the underlying costs and bandwidth constraints associated with maintaining up-to-date, coherent content across a particular network (e.g., network 480).

Three content routing techniques which may be selected (based on one or more of the foregoing variables) to maintain coherent content across the plurality of nodes illustrated in **Figure 4** are content revalidation, content notification, and content synchronization.

Content Revalidation

When content validation is selected, the original content source will be checked only when the content is requested locally. For example, client 473 may request an installation program for a new Web browser (e.g., the latest version of Microsoft's™ Internet Explorer™). The file may then be transmitted form ICDS server 460 to client 473 and a copy of the file cached locally on ICDS node 430. Consequently, if client 472 requests the same program, for example, two weeks later, ICDS node 430 may be configured to check ICDS server 460 to ensure that it contains the most recent copy of the file before passing it on to client 472 (i.e., ICDS node 430 "revalidates" the copy it has locally).

ICDS node 430 may also be configured to revalidate a piece of content only if has been stored locally for a predetermined amount of time (e.g., 1 week). The particular length of time selected may be based on one or more of the variables discussed above. Moreover, in one embodiment, the age/revision of a particular piece of content is determined based on tags (e.g., HTML metatags) inserted in the particular content/file.

Revalidation may work more efficiently with certain types of content than with others. For example, revalidation may be an appropriate mechanism for maintaining up-to-date copies of larger files which do not change very frequently (i.e., such as the program installation files described above). However, revalidation may not work as efficiently for caching smaller and/or continually changing files (e.g., small HTML files) because the step of revalidating may be just as time consuming as making a direct request to ICDS server 460 for the file itself. If the file in question is relatively small and/or is changing on a minute-by-minute basis (e.g., an HTML file containing stock quotes) then one or more other content routing techniques may be more appropriate.

Of course, other routing variables may influence the decision on which technique to use, including the issue of how strong the data transmission connection is between ICDS node 430 and ICDS server 460 (i.e., how reliable it is, how much bandwidth is available . . . etc) and the necessity that the underlying information cached locally (at ICDS node 430) be accurate. The important thing to remember is that ICDS node 430 – because of its underlying open API architecture – may be configured based on the unique preferences of a particular client.

Content Notification

Content notification is a mechanism wherein the central repository for a particular piece of content maintains a list of nodes, or "subscribers," which cache a copy of the content locally. For example, in **Figure 4**, a plurality of agents may run on ICDS server 460 which maintain a list of content subscribers (e.g., ICDS node 430, ICDS node 420 . . . etc) for specific types of content (e.g., HTML, data streaming files, FTP files . . . etc). In one embodiment of the system, a different agent may be executed for each protocol supported by ICDS server 460 and/or ICDS nodes 410-440.

When a particular piece of content is modified on ICDS server 460, a notification of the modification may be sent to all subscriber nodes (i.e., nodes which subscribe to that particular content). Upon receiving the notification, the subscriber node – e.g., ICDS node 430 – may then invalidate the copy of the content which it is storing locally. Accordingly, the next time the content is requested by a client (e.g., client 472), ICDS node 430 will retrieve the up-to-date copy of the content from ICDS server 460. The new copy will then be maintained locally on ICDS node 430 until ICDS node 430 receives a second notification from an agent running on ICDS server 460 indicating that a new copy exists.

Alternatively, each time content is modified on ICDS server 460 the modified content may be sent to all subscriber nodes along with the notification. In this manner a local, up-to-date copy of the content is always ensured. In one embodiment of the system, notification and/or transmittal of the updated content by the various system agents is done after a predetermined period of time has elapsed (e.g., update twice a day). The time period may be selected based on the importance of having an up-to-date copy across all nodes on the network 480, 490.

As was the case with content revalidation, the different varieties of content notification may work more efficiently in some situations than in others. Accordingly, content notification may be selected as a protocol (or not selected) based on one or more of the routing variables recited at the beginning of this section (i.e., the "content routing" section). For example, content notification may be an appropriate technique for content which is frequently requested at the various nodes across networks 480 and 490 (e.g., for the 5-10% of the content which is requested 90% of the time), but may be a less practical technique for larger amount of content which is requested infrequently. As another example, large files which change frequently may not be well suited for content notification (i.e., particularly the type of content notification where the actual file is sent to all subscribers along with the notification) due to bandwidth constraints across networks 480 and/or 490 (i.e., the continuous transmission of large, frequently changing files may create too much additional network traffic).

Content Synchronization

Content synchronization is a technique for maintaining an exact copy of a particular type of content on all nodes on which it is stored. Using content synchronization, as soon as a particular piece of content is modified at, for example, ICDS node 430, it will immediately be updated at all other nodes across networks 480 and/or 490. If the same piece of data was concurrently modified at one of its other storage locations (e.g., ICDS node 410) then the changes may be backed off in order to maintain data coherency. Alternatively, an attempt may be made to reconcile the two separate modifications if it is possible to do so (using, e.g., various data coherency techniques).

Once again, as with content notification and content revalidation, content synchronization is more suitable for some situations than it is for others. For example, content synchronization is particularly useful for information which can be modified from several different network nodes (by contrast, the typical content notification paradigm assumes that the content is modified at one central node). Moreover, content synchronization may be useful for maintaining content across a network which it is particularly important to keep current. For example, if network 480 is an automatic teller machine (hereinafter "ATM") network, then when a user withdraws cash from a first node (e.g., ICDS node 440), his account will be instantly updated on all nodes (e.g., ICDS node 410, 420, 460, and 430) to reflect the withdrawal. Accordingly, the user would not be able to go to a different node in a different part of the country and withdraw more than what he actually has in his account.

As another example, a user's account status on a network (i.e., whether he is a current subscriber and/or what his network privileges are) may be maintained using content synchronization. If, for example, a user of network 480 were arrested for breaking the law over network 480 (e.g., distributing child pornography), it would be important to disable his user account on all network nodes on which this information might be cached. Accordingly, using content synchronization, once his account was disabled at one node on network 480 this change would automatically be reflected across all nodes on the network.

As previously stated, the choice of which content routing technique to use for a particular type of content may be based on any of the variables set forth above. In one embodiment, the frequency with which content is accessed across the networks 480 and/or 490 may be an important factor in deciding which protocol to use. For example, the top 1% accessed content may be selected for content synchronization, the top 2%-10% accessed content may be selected for content notification, and the remaining content across networks 480, 490 may be selected for content revaildation.

III. Content Delivery Medium Selection

In addition to the content routing flexibility provided by the content delivery system as set forth above, one embodiment of the system allows content delivery nodes such as ICDS node 430 to select from a plurality of different transmission media. For example, ICDS node 430 may receive content from ICDS server 460 via a plurality of communication media, including, but not limited to, satellite transmission, wireless RF transmission, and terrestrial transmission (e.g., fiber).

Moreover, as with the selection of a particular content routing technique, the selection of a particular transmission medium may be based on any of the variables set forth above (see, e.g., routing variables listed under counter routing heading; page 24, line 18 through page 25, line 9). Moreover, the choice of a particular transmission medium may be dynamically adjustable based on performance of that medium. For example, ICDS node 430 may be configured to receive all of its content over terrestrial network 480 as long as network 480 is transmitting content at or above a threshold bandwidth. When transmissions over network 480 dip below the threshold bandwidth, ICDS node 480 may then begin receiving certain content via satellite broadcast or wireless communication.

In addition, a transmission medium may be selected for transmitting specific content based on how frequently that content is accessed. For example, the top 10% frequently accessed content may be continually pushed out to ICDS node 430 via satellite broadcast while the remaining content may be retrieved by (i.e., "pulled" to) ICDS node 430 over network 480 upon request by clients (e.g., client 473). Accordingly, those employing ICDS nodes such as node 430 can run a cost-benefit analysis to determine the most cost effective way to implement their system by taking in to consideration, for example, the needs of their users, the importance of the content involved and the expense of maintaining multiple transmission connections into ICDS node 430 (e.g., the cost associated with maintaining an ongoing satellite connection).

In one embodiment of the system, tags (e.g., HTML metatags) may be inserted into particular types of content to identify a specific transmission path/medium for delivering that content to ICDS node 480. The tags in this embodiment may identify to various nodes (and/or routers) across networks 480 and/or 490 how the particular content should be routed across the networks (e.g., from node 410 to node 420 via terrestrial network 480; from node 420 to node 430 via wireless transmission).

One of ordinary skill in the art will readily recognize from the preceding discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention. Throughout this detailed description, numerous specific details are set forth such as specific network protocols (i.e., RADIUS) and networks (i.e., the Internet) in order to provide a thorough understanding of the present invention. It will be appreciated by one having ordinary skill in the art, however, that the present invention may be practiced without such specific details. In other instances, well known software and hardware configurations/techniques have not been described in detail in order to avoid obscuring the subject matter of the present invention. The invention should, therefore, be measured in terms of the claims which follow.

CLAIMS

What is claimed is:

1. A network content delivery system configured to:

select a first content routing technique for processing a first set of network content; and

select a second content routing technique for processing a second set of network content, wherein said first and second content routing techniques are selected based on one or more content routing variables.

2. The network content delivery system as claimed in Claim 1 wherein one of said selected content routing techniques is a content revalidation technique.

3. The network content delivery system as claimed in Claim 1 wherein one of said selected content routing techniques is a content notification technique.

4. The network content delivery system as claimed in Claim 1 wherein one of said selected content routing techniques is a content synchronization technique.

5. The network content delivery system as claimed in Claim 1 wherein one of said content routing variables is the frequency with which said network content is accessed by users.

6. The network content delivery system as claimed in Claim 1 wherein one of said content routing variables is the size of said network content.

7. The network content delivery system as claimed in Claim 1 wherein one of said content routing variables is the frequency with which said network content is modified.

8. The network content delivery system as claimed in Claim 1 wherein one of said content routing variables is the type of network content (e.g., HTML, Usenet News).

9. The network content delivery system as claimed in Claim 1 wherein one of said content routing variables is identity of the user requesting said network content.

10. The network content delivery system as claimed in Claim 2 wherein said content revalidation technique is selected based on the size of said network content.

11. The network content delivery system as claimed in Claim 2 wherein said content revalidation technique is selected based on the frequency with which said network content is accessed.

12. The network content delivery system as claimed in Claim 3 wherein said content notification technique is selected based on the size of said network content.

13. The network content delivery system as claimed in Claim 3 wherein said content notification technique is selected based on the frequency with which said network content is accessed.

14. The network content delivery system as claimed in Claim 4 wherein said content synchronization technique is selected based on the size of said network content.

15. The network content delivery system as claimed in Claim 4 wherein said content synchronization technique is selected based on the frequency with which said network content is accessed.

16. The network content delivery system as claimed in Claim 1 including the additional step of selecting a first transmission medium for a first group of network content based on one or more of said content routing variables.

17. The network content delivery system as claimed in Claim 16 including the additional step of selecting a second transmission medium for a second group of network content based on one or more of said content routing variables.

18. The network content delivery system as claimed in Claim 1 including an application programming interface for interfacing with a plurality of network protocol and service modules.

19. A content delivery system comprising:

a network node for storing network content;

a first transmission medium communicatively coupled to said network node for transmitting a first set of network content to said network node; and

a second transmission medium communicatively coupled to said network node for transmitting a second set of network content to said network node,

wherein said first and second sets of network content are selected based on one or more routing variables.

20. The content delivery system as claimed in Claim 19 wherein said first transmission medium is a satellite transmission.

21. The content delivery system as claimed in Claim 19 wherein said first transmission medium is a wireless radio frequency transmission.

22. The content delivery system as claimed in Claim 19 wherein said first transmission medium is terrestrial-based transmission.

23. The content delivery system as claimed in Claim 19 wherein said network node monitors transmission bandwidth of said first transmission medium and reallocates content

from said first set to said second set if said first transmission medium drops below a predetermined threshold value.

24. The content delivery system as claimed in Claim 23 wherein said first transmission medium is terrestrial and said second transmission medium is non-terrestrial.

25. The content delivery system as claimed in Claim 19 wherein content is included in said first set based on the frequency with which said content is accessed.

26. The network content delivery system as claimed in Claim 19 including an application programming interface for interfacing with a plurality of network protocol and service modules.

27. An article of manufacture including a sequence of instructions stored on a computer-readable media which, when executed by a network node, cause the network node to perform the acts of:

establishing a plurality of groups of network content to be cached on said network node based on one or more content routing variables;

selecting a first content routing technique for maintaining data coherency in a first group of said plurality; and

selecting a second content routing technique for maintaining data coherency in a second group of said plurality.

28. The article of manufacture as claimed in claim 27 wherein said first content routing technique is content revalidation.

29. The article of manufacture as claimed in Claim 28 wherein said second content routing technique is content notification.

30. The article of manufacture as claimed in Claim 28 wherein said second content routing technique is content synchronization.

31. The article of manufacture as claimed in Claim 28 wherein said content routing variable used to select said content for said first group is the frequency with which said content is accessed.

32. The article of manufacture as claimed in Claim 29 wherein said content routing variable used to select said content for said first group is the frequency with which said content is accessed.

33. The article of manufacture as claimed in Claim 30 wherein said content routing variable used to select said content for said first group is the frequency with which said content is accessed.

34. A network node comprising:

an application programming interface ("API"), said API including a distributed relational database engine;

a plurality of protocol modules for interfacing with said API, said protocol modules configured to allow said system to communicate over a network using a plurality of network protocols;

a cache memory for caching data communicated to said cache memory using said plurality of protocol modules; and

a data services module for maintaining coherency between said data stored in said cache memory and data stored at other nodes across said network.

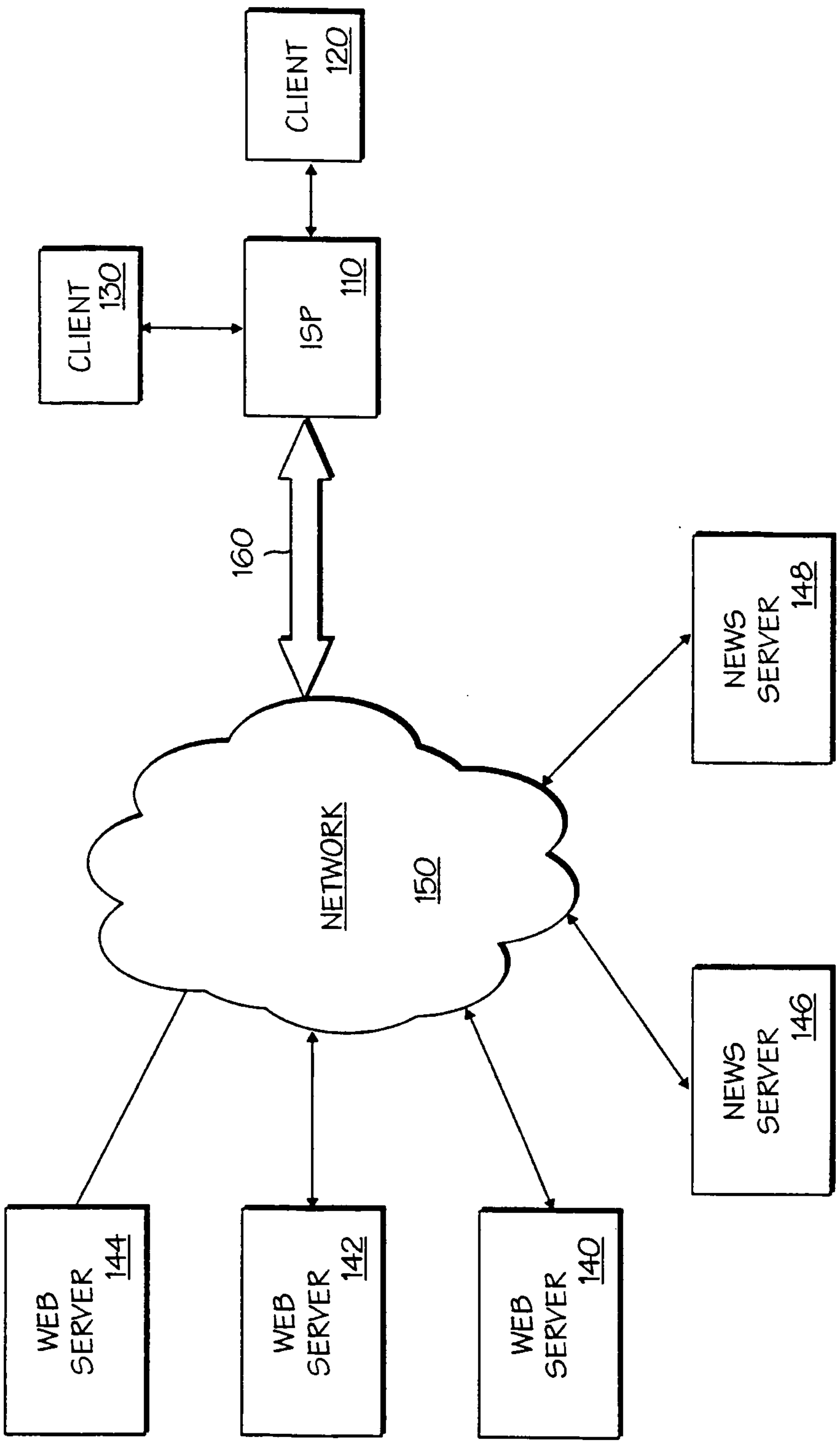

CLIENT 120
CLIENT 130
ISP 110
160
NETWORK 150
NEWS SERVER 148
NEWS SERVER 146
WEB SERVER 144
WEB SERVER 142
WEB SERVER 140

FIG. 1

2/4

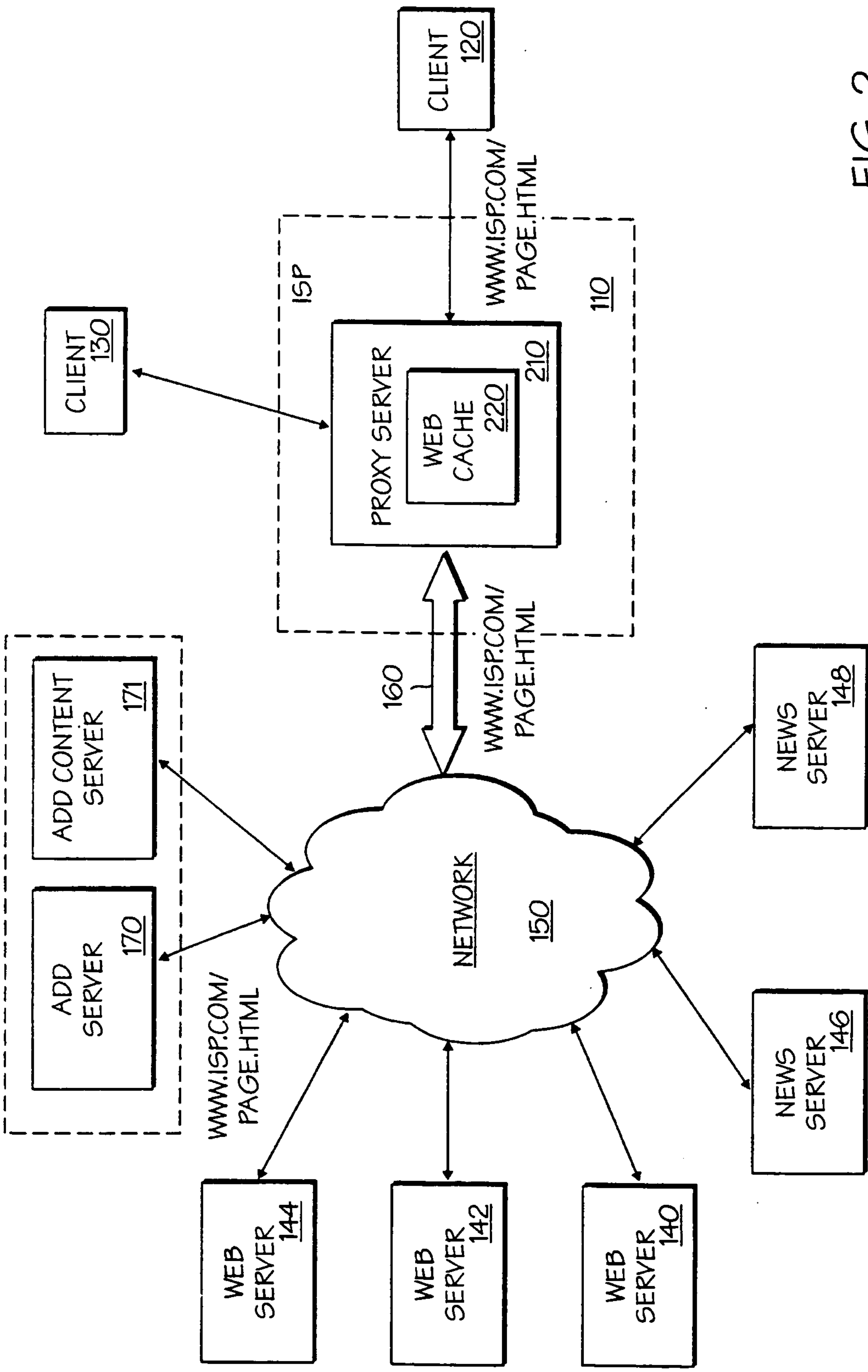

CLIENT 120
CLIENT 130
ISP
PROXY SERVER
WEB CACHE 220
210
110
WWW.ISP.COM/PAGE.HTML
WWW.ISP.COM/PAGE.HTML
160
ADD CONTENT SERVER 171
ADD SERVER 170
WWW.ISP.COM/PAGE.HTML
NETWORK
150
NEWS SERVER 148
NEWS SERVER 146
WEB SERVER 144
WEB SERVER 142
WEB SERVER 140

FIG. 2

3/4

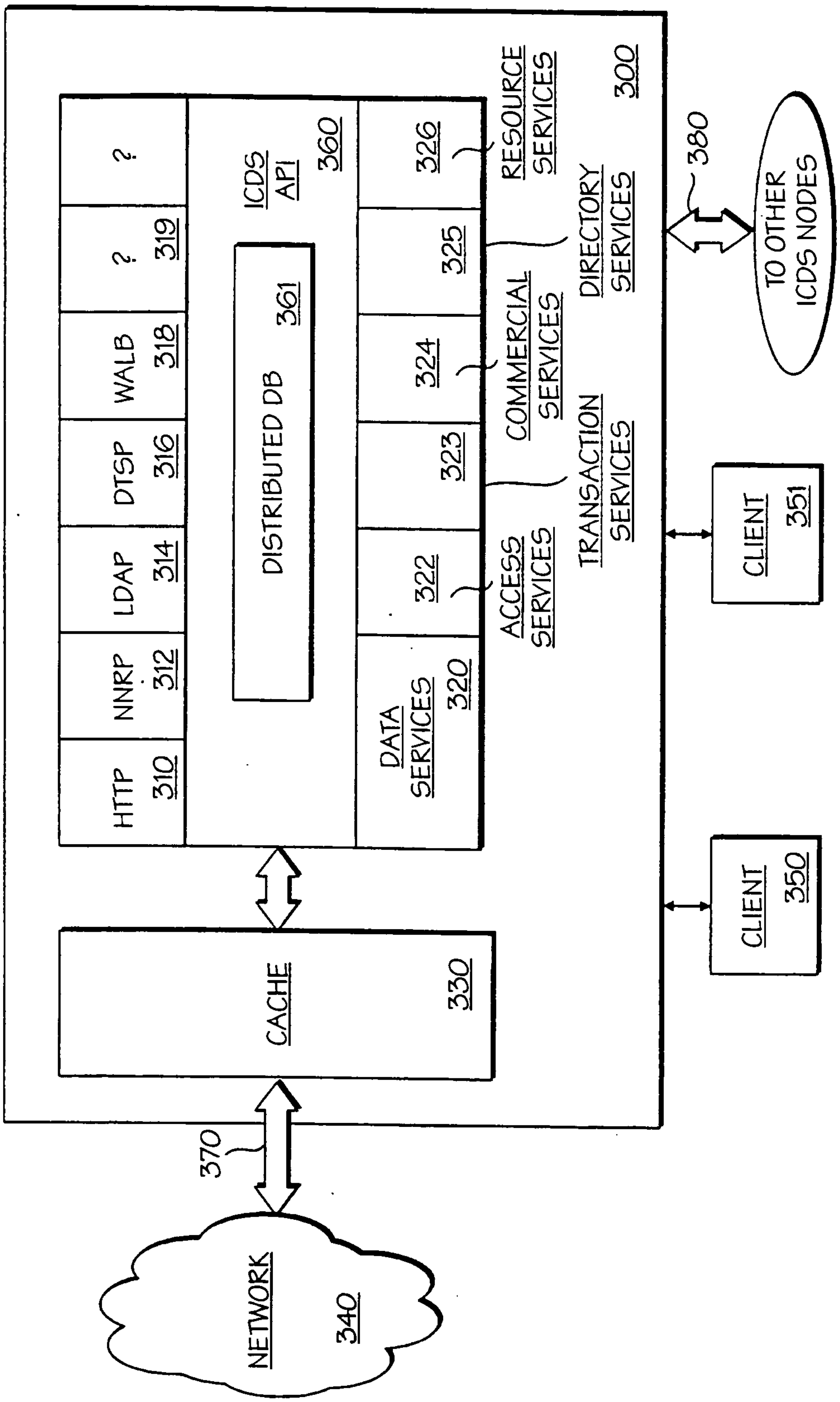

HTTP 310
NNRP 312
LDAP 314
DTSP 316
WALB 318
? 319
?
ICDS API 360
DISTRIBUTED DB 361
DATA SERVICES 320
322
323
324
325
326
ACCESS SERVICES
TRANSACTION SERVICES
COMMERCIAL SERVICES
DIRECTORY SERVICES
RESOURCE SERVICES
300
CACHE 330
370
NETWORK 340
CLIENT 350
CLIENT 351
380
TO OTHER ICDS NODES

FIG. 3

4/4

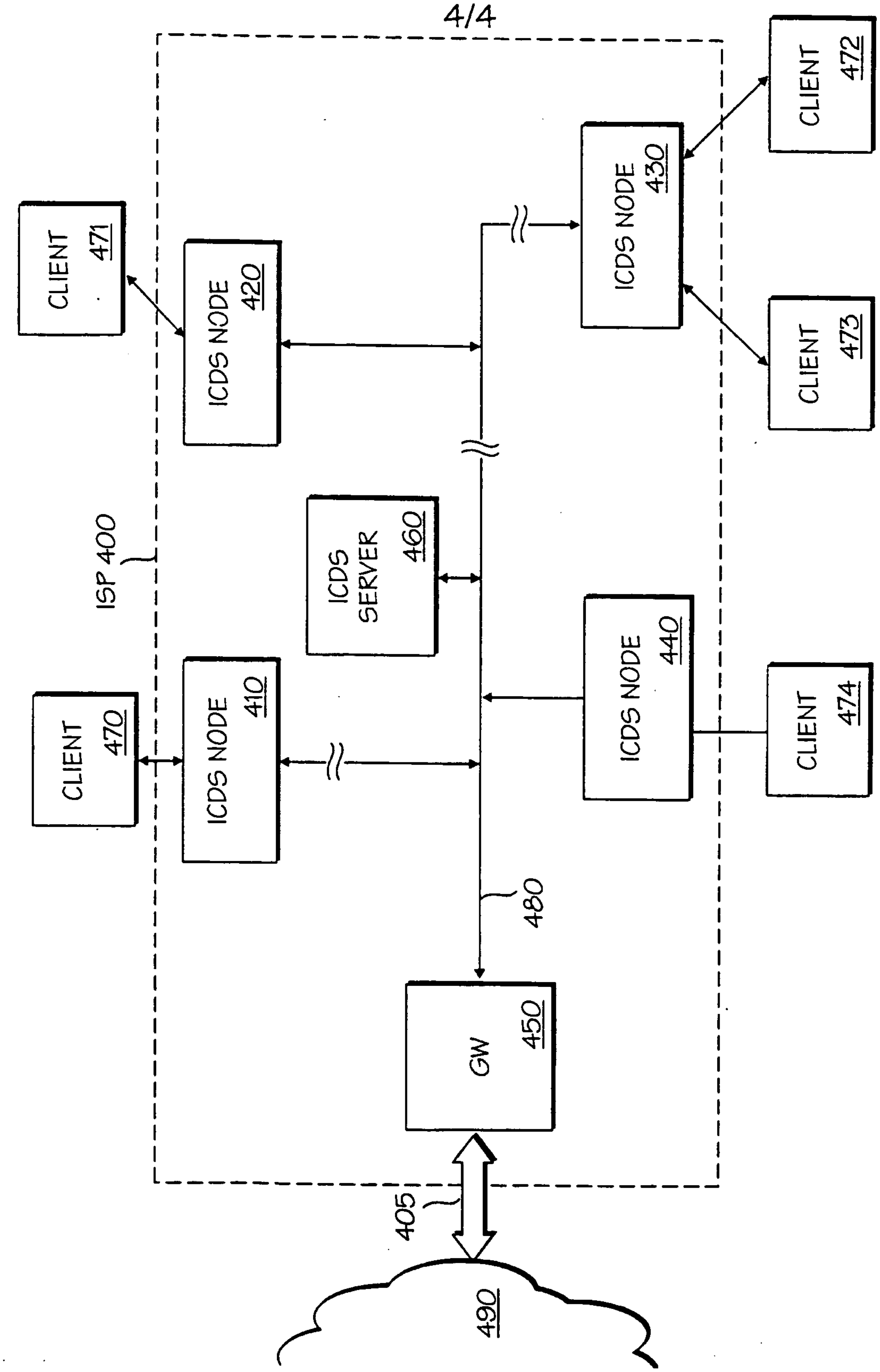

ISP 400
CLIENT 470
CLIENT 471
CLIENT 472
CLIENT 473
CLIENT 474
ICDS NODE 410
ICDS NODE 420
ICDS NODE 430
ICDS NODE 440
ICDS SERVER 460
GW 450
480
405
490

FIG. 4

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 May 2001 (25.05.2001)

PCT

(10) International Publication Number
WO 01/37209 A1

(51) International Patent Classification[7]: G06K 9/36, 9/46

(21) International Application Number: PCT/US00/30825

(22) International Filing Date: 8 November 2000 (08.11.2000)

(25) Filing Language: English

(26) Publication Language: Englisb

(30) Priority Data:
09/438,666 12 November 1999 (12.11.1999) US

(71) Applicant: TERALOGIC, INC. [US/US]; 1240 Villa Street, Mountain View, CA 94041 (US).

(72) Inventors: CASTOR, Jon, S.; 2160 Stockbridge Avenue, Woodside, CA 94062 (US). CHUI, Charles, K.; 340 Olive Street, Menlo Park, CA 94025 (US).

(74) Agents: WILLIAMS, Gary, S. et al.; Pennie & Edmonds LLP, 1155 Avenue of the Americas, New York, NY 10036 (US).

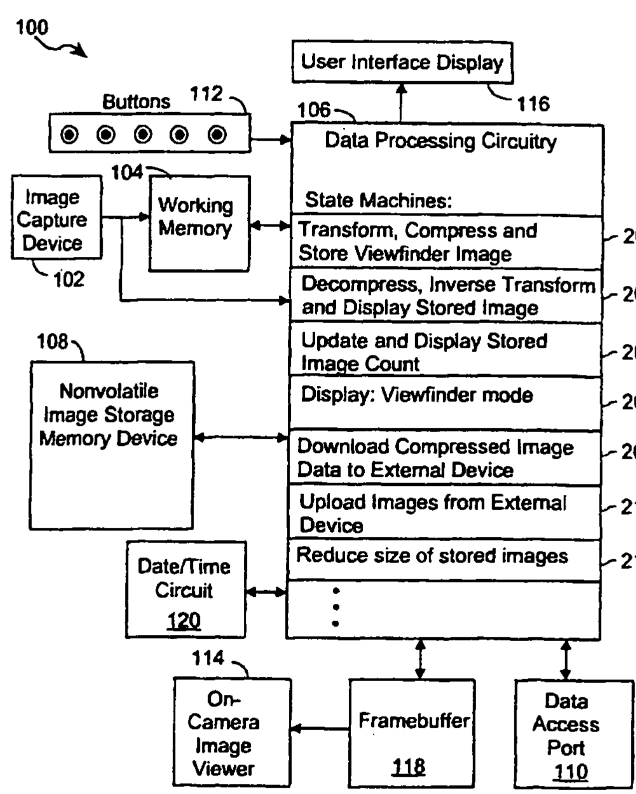
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, 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, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*

[Continued on next page]

(54) Title: PICTURE AND VIDEO STORAGE MANAGEMENT SYSTEM AND METHOD



(57) Abstract: An image processing system (100) stores image files in a memory device (108) at a number of incremental quality levels. Each image file has an associated image quality (that is fidelity or resolution) level corresponding to a quality level at which the corresponding image has been encoded. The images are initially encoded by applying a predefined transform, such as a DCT transform or wavelet-like transform (200), to image data received from an image capture mechanism (102) and then applying a data compression method to the transform data (200). The image is regenerated by successively applying a data decompression method and an inverse transform to an image file (202). Image file size reduction circuitry (212) and one or more state machines are used to lower the quality level of a specified one of the image files, including circuitry for extracting a subset of the data in the specified image file and forming a lower quality version of the specified image file that occupies less space in the memory device than was previously occupied by the specified image data structure. As a result, the amount of space occupied by image files in the memory device can be reduced so as to make room for the storage of additional image files or to allow more rapid transmission in a restricted bandwidth environment.

WO 01/37209 A1

— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

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.

PICTURE AND VIDEO STORAGE MANAGEMENT SYSTEM AND METHOD

The present invention relates generally to the storage of still and video images in an image processing device or system, such as a digital camera or digital video camera or a computer based image storage system, and particularly to a system and method for storing images at different image quality levels and for enabling users to dynamically compress high quality images into lower quality images in order to free up storage for the storage of additional images.

BACKGROUND OF THE INVENTION

Digital cameras typically include either permanent internal storage, and/or removable storage, for storing images. For instance, many digital cameras use removable flash memory cards for storing images. Each flash memory card has a fixed amount of memory, and when the memory card is full it can be removed from the camera and replaced by another flash memory card. In addition, digital cameras typically have a built-in viewer that enables the user to review the images stored on the flash memory card (and/or in the internal memory) and to delete user specified ones of the stored images. Deleting stored images obviously creates room for storing additional images.

When a digital camera user is "in the field" he/she generally has a limited amount of image storage on hand. If all the available image storage is full, the user has the choice of either not taking any additional pictures, or of deleting pictures from the image storage devices on hand to make room for new images. While this is actually one level better than the situation with film cameras, in which the user is simply out of luck when all the available film has been used, it is the premise of the present invention that the current image storage limitations of digital cameras are caused, in part, by failure to fully exploit the advantages of having images stored in digital format.

Similar storage vs. image quality considerations also apply to digitally encoded video frames. In particular, for any given amount of storage space, such as in a digital video camera, the goal is to retain the best image quality for the amount of storage required for a given number of video frames. Current devices allow the user to select image quality prior to capturing a digital video image, but do not enable the user to effectively manage the storage space in the video camera with respect to video sequences already taken, other than by deletion.

It is an object of the present invention to provide a digital camera or digital video camera, or other constrained storage device, that can store images at a plurality of image quality (i.e., fidelity or resolution) levels and furthermore can reduce images initially stored at a first image quality level to a lower image quality level so as to reduce the amount of storage occupied by those images.

It is also an object of the present invention to provide space efficient and computationally efficient image and video handling mechanisms for other applications, including network connected image and video libraries, Internet web browsers executed by client computers coupled to server computers, cable television set top boxes having video storage capabilities, and so on.

SUMMARY OF THE INVENTION

In summary, the present invention is an image processing device or system, such as a digital camera or digital video camera or a computer based image storage system, that can store images at a number of different image quality levels. The image processing device includes a memory device and image management logic. The memory device stores image files that each represent a respective image, each image file having an associated image quality level corresponding to a quality level at which the corresponding image has been encoded. The image management logic includes data processing circuitry and state machines for storing and processing image data received from an image capture mechanism. More specifically, the image management logic includes image processing circuitry and one or more state machines for applying a predefined transform, such as a wavelet-like transform, to image data received from the image capture mechanism to generate transform image data and for applying a data compression method to the transform image data so as to generate an image file having an associated image quality level.

The image management logic also includes image reconstruction circuitry and one or more state machines for successively applying a data decompression method and an inverse transform to a specified one of the image files so as to generate a reconstructed image suitable for display on an image viewer.

Further, the image management logic includes image file size reduction circuitry and one or more state machines for reducing the size of an image file while minimizing the reduction in image quality level. This circuitry extracts a subset of the data in the specified image file and forms a lower quality version of the specified image file that occupies less space in the memory device than was previously occupied by the specified image data structure. As a result, the amount of space occupied by image files in the memory device can be reduced so as to make room for the storage of additional image files or to allow more rapid transmission in a restricted bandwidth environment.

In a preferred embodiment, the image transform data in an image file is organized on a bit plane basis such that image transform data for at least one bit plane is stored in distinct portions of the image data structure from image transform data for other bit planes. To generate a lower quality image file, the image size reduction circuitry extracts a portion of the image file that excludes the image transform data for at least one bit plane and replaces the image file with an image file containing the extracted portion. Further, the image data is also organized on a transform layer basis such that image transform data for at least one transform layer is stored in distinct portions of the image data structure from image transform data for other transform layers. The image size reduction circuitry can also generate a lower quality image file by extracting a portion of the image file that excludes the image transform data for at least one transform layer and replaces the image file with an image file containing the extracted portion.

BRIEF DESCRIPTION OF THE DRAWINGS

Additional objects and features of the invention will be more readily apparent from the following detailed description and appended claims when taken in conjunction with the drawings, in which:

Fig. 1 is a block diagram of a digital camera in accordance with an embodiment of the present invention.

Fig. 2 depicts an image data array divided into a set of smaller analysis arrays for purposes of encoding and data compression.

Fig. 3 schematically depicts the process of transforming a raw image data array into a transform image array and compressing the transform image array into a compressed image file.

Figs. 4A and 4B depict image storage data structures.

Fig. 5 is a conceptual flow chart depicting changes in the state of a digital camera as various operations are performed.

Fig. 6 is a conceptual flow chart depicting changes in the state of a video image sequence as the video image sequence is encoded and compressed.

Figs. 7, 8 and 9 show data structures used in one particular embodiment for video image sequence encoding and compression.

Fig. 10 is a conceptual diagram of an Internet server and client devices that utilize the image or and video image compressing and management features of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In some image processing systems, an image can be stored at a number of discrete resolution levels, typically with each resolution level differing from its "neighbors" by a resolution factor of four. In other words, if the highest resolution representation (at resolution level 1) of the image contains X amount of information, the second resolution level representation contains (for example) X/4 amount of information, the third resolution level representation contains X/16 amount of information, and so on. Thus, an image's "resolution" typically means the amount of image information, and in the context of digital image processing systems is often expressed in terms of the number of distinct pixel elements. The number of resolution levels and the particular amount of information reduction from one level to the next may vary considerably from one system to another. Further, the present invention would be equally applicable to systems having a continuous range of resolution levels.

Another concept concerning image quality is "fidelity." The fidelity of an image can be compromised even if its resolution, in terms of the number of pixels in the image, is unchanged. An image's fidelity can be reduced by reducing the number of bits used to represent the image data. For instance, if the transform coefficients used to represent an image are represented at full fidelity using 12 bits, and then the number of bits used to represent transform coefficients is reduced to 11, the fidelity of the resulting image will be reduced. In other words, the quality of the image reconstructed from the lower fidelity data will be a little less sharp than an image reconstructed from higher fidelity data.

In this document, the terms "image quality" and "quality level" will be used in the general sense of image quality, encompassing both image "resolution" and image "fidelity." Thus, a reduction in an image's "image quality" from a top image quality level to a next highest image quality level might be accomplished either by reducing the image's resolution or by reducing its fidelity, or both. In other embodiments the terms "image quality" and "quality level" may be used to refer to other aspects of image quality as well.

Digital Camera Architecture

Referring to Fig. 1, there is shown an embodiment of a digital camera system 100 in accordance with the present invention. The digital camera system 100 includes an image capture device 102, such as a CCD or CMOS sensor array or any other mechanism suitable for capturing an image as an array of digitally encoded information. Thus the image capture device is assumed to include analog to digital conversion (ADC) circuitry for converting analog image information into digital values.

A working memory 104, typically random access memory, receives digitally encoded image information from the image capture device 102. More generally, it is used to store a digitally encoded image while the image is being transformed and compressed and otherwise processed by the camera's data processing circuitry 106. Memory 104 may be integrated on the same integrated circuit as other devices, or may be implemented using separate circuit(s).

The data processing circuitry 106 in one embodiment consists of hardwired logic and a set of state machines for performing a set of predefined image processing operations. In alternate embodiments the data processing circuitry 106 could be implemented in part or entirely using a fast general purpose microprocessor and a set of software procedures. However, at least using the technology available in 1999, it would be difficult to process and store full resolution images (e.g., full color images having 1280 x 840 pixels) fast enough to enable the camera to be able to take, say, twenty pictures per second, which is a requirement for some commercial cameras, as well as digital video cameras. In the future, general purpose microprocessors or general purpose image data microprocessors (e.g., single instruction multiple data (SIMD) processors) may be able to provide the fast image processing needed by digital cameras, in which case the data processing circuit 106 could be implemented using such a general purpose microprocessor or perhaps a hybrid processor system.

Each image, after it has been processed by the data processing circuitry 106, is typically stored as an "image file" in a nonvolatile memory storage device 108, typically implemented using "flash" (i.e., EEPROM) memory technology. The nonvolatile memory storage device 108 is preferably implemented as a removable memory card. This allows the camera's user to remove one memory card, plug in another, and then take additional pictures. However, in some implementations, the nonvolatile memory storage device 108 may not be removable, in which case the camera will typically have a data access port 110 to enable the camera to transfer image files to and from other devices, such as general purpose, desktop computers, computer systems and devices used to warehouse libraries of images, computer systems and devices used to store and distribute image files, and so on. Digital cameras with removable nonvolatile memory 108 may also include a data access port 110.

While the amount of storage in the nonvolatile image memory 108 will vary from one implementation to the next, such devices will typically have sufficient capacity to store 10 to 50 high quality images. Once the nonvolatile image memory 108 is full, if the file size reduction methodology of the present invention is not used, the only way the camera can be used to take additional pictures is either by deleting images from the nonvolatile image memory 108 or by replacing the nonvolatile image memory 108 with another one. If neither of these options are feasible (e.g., because the user has filled all the memory cards he/she has on hand with images that he/she does not wish to delete), then no further pictures can be taken until a new memory device 108 is inserted or the stored images are transferred to an external device (if available).

The digital camera 100 includes a set of buttons 112 for giving commands to the camera. In addition to the image capture button, there will typically be several other buttons to enable the use to select the quality level of the next picture to be taken, to scroll through the images in memory for viewing on the camera's image viewer 114, to delete images from the nonvolatile image memory 108, and to invoke all the camera's other functions. Such other functions might include enabling the use of a flash light source, and transferring image files to and from a computer. In accordance with the present invention, the user selectable functions, selected by using the buttons 112, further include reducing the size of one or more of the image files stored in the nonvolatile image memory 108 so as to make room for the storage of additional images. The buttons in one embodiment are electromechanical contact switches, but in other embodiments at least some of the buttons may be implemented as touch screen buttons on a user interface display 116, or on the image viewer 114.

The user interface display 116 is typically implemented either (A) as an LCD display device separate from the image viewer 114, or (B) as images displayed on the image viewer 114. Menus, user prompts, and information about the images stored in the nonvolatile image memory 108 may be displayed on the user interface display 116, regardless of how that display is implemented.

After an image has been captured, processed and stored in nonvolatile image memory 108, the associated image file may be retrieved from the memory 108 for viewing on the image viewer. More specifically, the image file is converted from its transformed, compressed form back into a data array suitable for storage in a framebuffer 118. The image data in the framebuffer is displayed on the image viewer 114. A date/time circuit 120 is used to keep track of the current date and time, and each stored image is typically date stamped with the date and time that the image was taken.

Image Data Structures

Referring to Fig. 2, in one embodiment the nonvolatile image memory 108 stores a directory 130 that lists all the image files 132 stored in the memory 108. Preferably, the directory 130 contains information for each stored image file 132, such as the date and time the image was taken, the quality level of the image and the file's location in the memory 108.

To understand the image data structure stored in each image file, it is helpful to first understand how an image file is encoded. Referring to Fig. 3, a raw image data array 140, obtained from the digital camera's image capture mechanism 102 (Fig. 1), is treated as a set of non-overlapping "analysis arrays" 142 of a fixed size, such as 32 x 32, or 64 x 64 (or more generally $2^n \times 2^n$, for some integer value of n). A sufficient number of subarrays are used to cover the entire data array that is to be encoded, even if some of the subarrays overhang the edges of the data array. The overhanging portions of the subarrays are filled with zero data values during the data encoding process. In a preferred embodiment, the origin of the data array is the top left corner, the first coordinate used to identify data array positions is the "Y" axis or vertical coordinate, and the second coordinate used is the "X" axis or horizontal coordinate. Thus, a position of 0,64 indicates a pixel at the top vertical position of the array, 64 pixel positions over to the right from the array origin, while a position of 32,0 indicates a pixel on the left edge of the array, 32 pixel positions vertically down from the array origin.

An appropriate transform is applied to each of the analysis arrays 142, and then the resulting transform coefficients for each analysis array are quantized (e.g., divided by an appropriate value to generate integer valued, quantized transform coefficients) so as to generate a transformed image array 144. In one embodiment the transform applied to the raw image data is a wavelet-like transform. In other embodiments a DCT transform could be used (which is the type of transform used in current JPEG image encoding systems), or other types of wavelet or wavelet-like transforms could be used.

In this document, the terms "wavelet" and "wavelet-like" are used interchangeably. Wavelet-like transforms generally have spatial frequence characteristics similar to those of conventional wavelet transforms, and are losslessly reversible, but have shorter filters that are more computationally efficient.

In one embodiment the transformed image array 144 is generated by successive applications of a wavelet-like decomposition transform. A first application of the wavelet-like decomposition transform to an initial two dimensional array of "raw" image data generates four sets of coefficients, labeled LL, HL1, LH1 and HH1. Each succeeding application of the wavelet-like decomposition transform is applied only to the LL set of coefficients generated by the previous wavelet transformation step and generates four new sets of coefficients, labeled LL, HLx, LHx and HHx, where x represents the wavelet transform "layer" or iteration. After the last wavelet-like decomposition transform iteration only one LL set remains. The total number of coefficients generated is equal to the number of data samples in the original data array. The different sets of coefficients generated by each transform iteration are sometimes called layers. The number of wavelet transform layers generated for an image is typically a function of the resolution of the initial image. Performing five to seven wavelet transformation layers is fairly typical, but more or less may be used depending on such considerations as the size of the analysis arrays, the subject matter of the image, the data processing resources available for image compression, and the like.

For the purposes of explaining the operation of the image encoding and decoding operations of the present invention, the specific type of image transform used and the specific type of data quantization used to transform a raw image file 140 into a transformed image array 142 are not relevant and therefore are not further described herein. However, a preferred embodiment of the wavelet transform and data quantization methods are described in U.S. Patent No. 5,909518, "System and Method for Performing Wavelet and Inverse Wavelet Like Transformations of Digital Data," which is hereby incorporated by reference as background information.

Each transformed image array 144 is compressed and encoded using a sparse data encoding technique. In one embodiment, the method of compressing and encoding the analysis arrays is the method described in detail in U.S. patent application 08/858,035, filed May 16, 1997, entitled "System and Method for Scalable Coding of Sparse Data Sets," now U.S. Patent No. 5,949,911, which is hereby incorporated by reference as background information. The encoded image data for all the analysis arrays of the image are combined and stored as an image file 132.

Referring to Fig. 4A, the image file 132 includes header data 160 and a sequence of data structures 162, each representing one analysis array. The header data 160 indicates the size of the image file and the image file's quality level. The header data also includes a list of analysis array size values indicating the length of each of the analysis array data structures 162, thereby enabling fast indexing into the image data. Storing size values for the analysis arrays enables the camera's data processing circuitry 106 (Fig. 1) to locate the beginning of any analysis array data structure 162 without having to decode the contents of the earlier analysis arrays in the image file 132.

As shown in Fig. 4B, the encoded data 162 representing any one analysis array is stored in "bit layer order". For each analysis array, the encoding procedure determines the most significant non-zero bit in the data to be encoded, which is herein called the y^{th} bit. The value of y is determined by computing the maximum number of bits required to encode the absolute value of any data value in the analysis array. In particular, y is equal to $\mathrm{int}(\log_2 V) + 1$, where V is the largest absolute value of any element in the analysis array, and "int()" represents the integer portion of a specified value.

The encoded data 162 representing one analysis array includes (A) header data 170 indicating the maximum number of bits required to encode the absolute value of any data value in the analysis array, and (B) a sequence of data structures 172, each representing one bit plane of the elements in the analysis array. The x^{th} bit plane of the analysis array is the x^{th} bit of the absolute value of each of the elements in the analysis array. A sparse data encoding technique is used so that it takes very little data to represent a bit plane that contains mostly zero values. Typically, higher frequency portions of the transformed, quantized image data will contain more zero values than non-zero values, and further most of the non-zero values will have relatively small absolute value. Therefore, the higher level bit planes of many analysis arrays will be populated with very few non-zero bit values.

In an alternate embodiment, the data structure shown in Fig. 4A is modified slightly. In particular, to facilitate fast extraction of lower-resolution image data from an image file, the boundaries of the analysis arrays are adjusted, if necessary, so as to coincide precisely with the boundaries between the wavelet transform regions shown in Fig. 3 (e.g., the boundary between HL2 and HL1). If the size of the initial image array is not equal to an integer number of analysis arrays (i.e., if either the height or width of the image array is not an integer multiple of 2^n, where the size of each analysis array is 2^n x 2^n for an integer value of n), at least some of the boundaries between wavelet transform regions will fall in the middle of the analysis regions. For example, for a 800 x 600 pixel image, the LL region might have a size of 50 x 38. If the wavelet transform coefficients are encoded in units of analysis regions of size 32 x 32, the LL region will be encoded in four analysis regions, three of which would normally contain data for neighboring wavelet transform regions. In this alternate embodiment, each analysis array that overlaps a border between wavelet transform regions is replaced by two or four analysis regions (depending on whether the analysis array overlaps one or two region boundaries), with zero values being stored in the appropriate locations so that each analysis array contains data from only one wavelet transform region. The analysis arrays are still stored in "origin sorted order" in the image file 132, with the "origin" now being defined as the coordinate of the coefficient closest to the upper left corner of the analysis array that has not been overwritten with zero values.

In another alternate embodiment, a different transform than the wavelet-like transform could be used, but the resulting image data would still be stored in bit plane order. For instance, a DCT transform could be used.

In some embodiments of the present invention, the raw image array received from the digital camera's image capture mechanism may first be divided into "analysis arrays" and then transformed and quantized. Further, the analysis arrays may each be a thin horizontal strip of the image array. That is, each analysis array may extend the full width of the image array, but have a height of only a few (e.g., 4 to 16) image elements. In yet another embodiment, the image array might not be divided into analysis arrays at all.

Generally, in all embodiments described above, the compressed encoded image data is stored in bit plane order. The reason that bit plane ordered storage is favored is that it makes gradual fidelity reduction very easy: to reduce the fidelity of an image file by a minimum amount, the data for the lowest level bit plane in the file is discarded and the remaining image data is retained, resulting in a smaller file with one bit plane less fidelity.