4:18-34	2:64 to 3:12
4:35 to 5:22	3:13-36
5:23-34 - Blank	
5:25 "SUMMARY OF THE INVENTION"	
5:26-34	7:22-29
5:35 to 6:9	7:30-49
6:10 - Blank	
6:11 "DETAILED DESCRIPTION"	
6:12 to 7:18	11:17-55
7:19-36	11:56 to 12:4
8:1-24	12:5-25
8:25 to 9:21	12:26-55
9:22 to 10:6	12:56 to 13:8
10:7-11:8	13:9-44
11:9-24	13:45-59
11:25-34	13:60-67
11:35 to 12:2	14:1-6
12:3-35	14:7-34
12:36 to 13:22	14:35-56
13:23-30	14:57-64
13:31 to 14:34	14:65 to 15:35
14:35 to 15:17	15:36-53
15:18-26	15:54-61
(DISCLOSURE ENDS AT 15:26)	

IV. Jurat

22. I have been warned that willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. 1001) and may jeopardize the validity of the application or any

patent issuing thereon. All statements I make in the declaration I either know to be true or on information and belief I believe them to be true.

Signed:

SCOTT MOSKOWITZ

Attachment 11 Page 16 of 16

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE

PATENT APPLICATION TRANSMITTAL LETTER	ATTORNEY DOCKET NUMBER: **1607/6**

Assistant Commissioner for Patents
Washington D.C. 20231

Transmitted herewith for filing is the patent application of

Inventor(s) : Scott A. Moskowitz

For : METHOD FOR COMBINING TRANSFER FUNCTIONS WITH PREDETERMINED KEY CREATION

Enclosed are:

1. **17** sheets of specification, **8** sheets of claims, and **1** sheet of abstract.
2. **1** sheet(s) of drawings.
3. Executed Declaration/Power of Attorney.
4. Executed Verified Statement (Declaration) Claiming Small Entity Status -Independent Inventor.

The filing fee has been calculated as shown below:

	NUMBER FILED		NUMBER EXTRA*	RATE ($)	FEE ($)
BASIC FEE				790.00	$ 790.00
TOTAL CLAIMS	29	- 20 =	9	22.00	$ 198.00
INDEPENDENT CLAIMS	6	- 3 =	3	82.00	$ 246.00
MULTIPLE DEPENDENT CLAIM PRESENT				270.00	
FEE FOR RECORDATION OF ASSIGNMENT				40.00	
* Number extra must be zero or larger				TOTAL	$ 1,234.00
If applicant is a small entity under 37 C.F.R. §§ 1.9 and 1.27, then divide total fee by 2, and enter amount here.				SMALL ENTITY TOTAL	$ 617.00

5. The Office is authorized to charge the filing fee of $ 617.00 to Deposit Account No. 11-0600. The Office is further authorized to charge any additional fees or credit any overpayments to the above deposit account number. A copy of this letter is being submitted to facilitate processing of this application.

Dated: March 24, 1998

Patrick Buckley
Patrick J. Buckley (Reg. No. 40,928)

KENYON & KENYON
1025 Connecticut Avenue, N.W.
Suite 600
Washington, D.C. 20036-5405
(202) 429-1776 (202) 429-0796 (fax)

157250

Attachment 13 Page 1 of 31

Docket No.: 1607/6

METHOD FOR COMBINING TRANSFER FUNCTIONS WITH PREDETERMINED KEY CREATION

FIELD OF THE INVENTION

The invention relates to the protection of digital information. More particularly, the invention relates to a method for combining transfer functions with predetermined key creation.

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. patent application Serial No. 08/587,943, filed January 17, 1996, entitled "Method for Stega-Cipher Protection of Computer Code," the entire disclosure of which is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

Increasingly, commercially valuable information is being created and stored in "digital" form. For example, music, photographs and video can all be stored and transmitted as a series of numbers, such as 1's and 0's. Digital techniques let the original information be recreated in a very accurate manner. Unfortunately, digital techniques also let the information be easily copied without the information owner's permission.

Because unauthorized copying is clearly a disincentive to the digital distribution of valuable information, it is important to establish responsibility for copies and derivative copies of such works. For example, if each authorized digital copy of a popular song is identified with a unique number, any unauthorized copy of the song would also contain the number. This would allow the owner of the information, such as a song publisher, to investigate who made the unauthorized copy. Unfortunately, it is possible that the unique number could be erased or altered if it is simply tacked on at the beginning or end of the digital information.

As will be described, known digital "watermark" techniques give creators and publishers of digitized multimedia content localized, secured identification and authentication of that content. In considering the various forms of multimedia content, such as "master," stereo, National Television Standards Committee (NTSC) video, audio tape or compact disc, tolerance of quality will vary with individuals and affect the underlying commercial and aesthetic value of the content. For example, if a digital version of a popular song sounds distorted, it will be less valuable to users. It is therefore desirable to embed copyright, ownership or purchaser information, or some combination of these and related data, into the content in a way that will damage the content if the watermark is removed without authorization.

To achieve these goals, digital watermark systems insert ownership information in a way that causes little or no noticeable effects, or "artifacts," in the underlying content signal. For example, if a digital watermark is inserted into a digital version of a song, it is important that a listener not be bothered by the slight changes introduced by the watermark. It is also important for the watermark technique to maximize the encoding level and "location sensitivity" in the signal to force damage to the content signal when removal is attempted. Digital watermarks address many of these concerns, and research in the field has provided extremely robust and secure implementations.

What has been overlooked in many applications described in the art, however, are systems which closely mimic distribution of content as it occurs in the real world. For instance, many watermarking systems require the original un-watermarked content signal to enable detection or decode operations. These include highly publicized efforts by NEC, Digimarc and others. Such techniques are problematic because, in the real world, original master copies reside in a rights holders vaults and are not readily available to the public.

With much activity overly focused on watermark survivability, the security of a digital watermark is suspect. Any simple linear operation for encoding information into a signal may be used to erase the embedded signal by inverting the process. This is not a difficult task, especially when detection software is a plug-in freely available to the public, such as with Digimarc. In general, these systems seek to embed cryptographic information, not cryptographically embed information into target media content.

Other methods embed ownership information that is plainly visible in the media signal, such as the method described in US Patent No. 5,530,739 to Braudaway et al. The system described in Braudaway protects a digitized image by encoding a visible watermark to deter piracy. Such an implementation creates an immediate weakness in securing the embedded information because the watermark is plainly visible. Thus, no search for the embedded signal is necessary and the watermark can be more easily removed or altered. For example, while certainly useful to some rights owners, simply placing the symbol "©" in the digital information would only provide limited protection. Removal by adjusting the brightness of the pixels forming the "©" would not be difficult with respect to the computational resources required.

Other relevant prior art includes US Patents No. 4,979,210 and 5,073,925 to Nagata et al., which encodes information by modulating an audio signal in the amplitude/time domain. The modulations introduced in the Nagata process carry a "copy/don't copy" message, which is easily found and circumvented by one skilled in the art. The granularity of encoding is fixed by the amplitude and frequency modulation limits required to maintain inaudibility. These limits are relatively low, making it impractical to encode more information using the Nagata process.

Although US Patent No. 5,664,018 to Leighton describes a means to prevent collusion attacks in digital watermarks, the disclosed method may not actually provide the security described. For example, in cases where the watermarking technique is linear, the "insertion envelope" or "watermarking space" is well-defined and thus susceptible to attacks less sophisticated than collusion by unauthorized parties. Over-encoding at the watermarking encoding level is but one simple attack in such linear implementations. Another consideration not made by Leighton is that commercially-valuable content may already exist in a un-watermarked form somewhere, easily accessible to potential pirates, gutting the need for any type of collusive activity. Digitally signing the embedded signal with preprocessing of watermark data is more likely to prevent successful collusion. Furthermore, a "baseline" watermark as disclosed is quite subjective. It is simply described elsewhere in the art as the "perceptually significant" regions of a signal. Making a watermarking function less linear or inverting the insertion of watermarks would seem to provide the same benefit without the additional work required to create a "baseline" watermark. Indeed, watermarking algorithms should already be capable of defining a target insertion envelope or region without additional steps. What is evident is the Leighton patent does not allow for initial prevention of attacks on an embedded watermark as the content is visibly or audibly unchanged.

It is also important that any method for providing security also function with broadcasting media over networks such as the Internet, which is also referred to as "streaming." Commercial "plug-in" products such as RealAudio and RealVideo, as well as applications by vendors VDONet and Xtreme, are common in such network environments. Most digital watermark implementations focus on common file base signals and fail to anticipate the security of streamed signals. It is desirable that any protection scheme be able to function with a plug-in player without advanced knowledge of the encoded media stream.

Other technologies focus solely on file-based security. These technologies illustrate the varying applications for security that must be evaluated for different media and distribution environments. Use of cryptolopes or cryptographic containers, as proposed by IBM in its Cryptolope product, and InterTrust, as described in U.S. Patents No. 4,827,508, 4,977,594, 5,050,213 and 5,410,598, may discourage certain forms of piracy. Cryptographic containers, however, require a user to subscribe to particular decryption software to decrypt data. IBM's InfoMarket and InterTrust's DigiBox, among other implementations, provide a generalized model and need proprietary architecture to function. Every user must have a subscription or registration with the party which encrypts the data. Again, as a form of general encryption, the data is scrambled or encrypted without regard to the media and its formatting. Finally, control over copyrights or other neighboring rights is left with the implementing party, in this case, IBM, InterTrust or a similar provider.

Methods similar to these "trusted systems" exist, and Cerberus Central Limited and Liquid Audio, among a number of companies, offer systems which may functionally be thought of as subsets of IBM and InterTrust's more generalized security offerings. Both Cerberus and Liquid Audio propose proprietary player software which is registered to the user and "locked" in a manner parallel to the locking of content that is distributed via a cryptographic container. The economic trade-off in this model is that users are required to use each respective companies' proprietary player to play or otherwise manipulate content that is downloaded. If, as is the case presently, most music or other media is not available via these proprietary players and more companies propose non-compatible player formats, the proliferation of players will continue. Cerberus and Liquid Audio also by way of extension of their architectures provide for "near-CD quality" but proprietary compression. This requirement stems from the necessity not to allow content that has near-identical data make-up to an existing consumer electronic standard, in Cerberus and Liquid Audio's case the so-called Red Book audio CD standard of 16 bit 44.1 kHz, so that comparisons with the proprietary file may not yield how the player is secured. Knowledge of the player's file format renders its security ineffective as a file may be replicated and played on any common player, not the intended proprietary player of the provider of previously secured and uniquely formatted content. This is the parallel weakness to public key crypto-systems which have gutted security if enough plain text and cipher text comparisons enable a pirate to determine the user's private key.

Many approaches to digital watermarking leave detection and decoding control with the implementing party of the digital watermark, not the creator of the work to be protected. A set of secure digital watermark implementations address this fundamental control issue forming the basis of key-based approaches. These are covered by the following patents and pending applications, the entire disclosures of which are hereby incorporated by reference: US Patent No. 5,613, 004 entitled "Steganographic Method and Device" and its derivative US patent application Serial No. 08/775,216, US patent application Serial No. 08/587,944 entitled "Human Assisted Random Key Generation and Application for Digital Watermark System," US Patent Application Serial No. 08/587,943 entitled "Method for Stega-Cipher Protection of Computer Code," US patent application Serial No. 08/677,435 entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data," and US Patent Application Serial No. 08/772,222 entitled "Z-Transform Implementation of Digital Watermarks." Public key crypto-systems are described in US Patents No. 4,200,770, 4,218,582, 4,405,829 and 4,424,414, the entire disclosures of which are also hereby incorporated by reference.

In particular, an improved protection scheme is described in "Method for Stega-Cipher Protection of Computer Code," US patent application Serial No. 08/587,943. This technique uses the key-based insertion of binary executable computer code within a content signal that is subsequently, and necessarily, used to play or otherwise manipulate the signal in which it is encoded. With this system, however, certain computational requirements, such as one digital player per digital copy of content, may be necessitated. For instance, a consumer may download many copies of watermarked content. With this technique, the user would also be downloading as many copies of the digital player program. While this form of security may be desirable for some applications, it is not appropriate in many circumstances.

Finally, even when digital information is distributed in encoded form, it may be desirable to allow unauthorized users to play the information with a digital player, perhaps with a reduced level of quality. For example, a popular song may be encoded and freely distributed in encoded form to the public. The public, perhaps using commonly available plug-in digital players, could play the encoded content and hear the music in some degraded form. The music may sound choppy, or fuzzy or be degraded in some other way. This lets the public decide, based on the available lower quality version of the song, if they want to purchase a key from the publisher to decode, or "clean-up," the content. Similar approaches could be used to distribute blurry pictures or low quality video. Or even "degraded" text, in the sense that only authenticated portions of the text can be determined with the predetermined key or a validated digital signature for the intended message.

In view of the foregoing, it can be appreciated that a substantial need exists for a method allowing encoded content to be played, with degraded quality, by a plug-in digital player, and solving the other problems discussed above.

SUMMARY OF THE INVENTION

The disadvantages of the art are alleviated to a great extent by a method for combining transfer functions with predetermined key creation. In one embodiment, digital information, including a digital sample and format information, is protected by identifying and encoding a portion of the format information. Encoded digital information, including the digital sample and the encoded format information, is generated to protect the original digital information.

In another embodiment, a digital signal, including digital samples in a file format having an inherent granularity, is protected by creating a predetermined key. The predetermined key is comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

With these and other advantages and features of the invention that will become hereinafter apparent, the nature of the invention may be more clearly understood by reference to the following detailed description of the invention, the appended claims and to the several drawings attached herein.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block flow diagram of a method for copy protection or authentication of digital information according to an embodiment of the present invention.

DETAILED DESCRIPTION

In accordance with an embodiment of the present invention, a method combines transfer functions with predetermined key creation. Increased security is achieved in the method by combining elements of "public-key steganography" with cryptographic protocols, which keep in-transit data secure by scrambling the data with "keys" in a manner that is not apparent to those with access to the content to be distributed. Because different forms of randomness are combined to offer robust, distributed security, the present invention addresses an architectural "gray space" between two important areas of security: digital watermarks, a subset of the more general art of steganography, and cryptography. One form of randomness exists in the mask sets that are randomly created to map watermark data into an otherwise unrelated digital signal. The second form of randomness is the random permutations of data formats used with digital players to manipulate the content with the predetermined keys. These forms can be thought of as the transfer function versus the mapping function inherent to digital watermarking processes.

According to an embodiment of the present invention, a predetermined, or randomly generated, key is used to scramble digital information in a way that is unlike known "digital watermark" techniques and public key crypto-systems. As used herein, a key is also referred to as a "mask set" which includes one or more random or pseudo-random series of bits. Prior to encoding, a mask can be generated by any cryptographically secure random generation process. A block cipher, such as a Data Encryption Standard (DES) algorithm, in combination with a sufficiently random seed value, such as one created using a Message Digest 5 (MD5) algorithm, emulates a cryptographically secure random bit generator. The keys are saved in a database, along with information matching them to the digital signal, for use in descrambling and subsequent viewing or playback. Additional file format or transfer property information is prepared and made available to the encoder, in a bit addressable manner. As well, any authenticating function can be combined, such as Digital Signature Standard (DSS) or Secure Hash Algorithm (SHA).

Using the predetermined key comprised of a transfer function-based mask set, the data representing the original content is manipulated at the inherent granularity of the file format of the underlying digitized samples. Instead of providing, or otherwise distributing, watermarked content that is not noticeably altered, a partially "scrambled" copy of the content is distributed. The key is necessary both to register the sought-after content and to descramble the content into its original form.

The present invention uses methods disclosed in "Method for Stega-Cipher Protection of Computer Code," US Patent Application Serial No. 08/587,943, with respect to transfer functions related to the common file formats, such as PICT, TIFF, AIFF, WAV, etc. Additionally, in cases where the content has not been altered beyond being encoded with such functional data, it is possible for a digital player to still play the content because the file format has not been altered. Thus, the encoded content could still be played by a plug-in digital player as discrete, digitally sampled signals, watermarked or not. That is, the structure of the file can remain basically unchanged by the watermarking process, letting common file format based players work with the "scrambled" content.

For example, the Compact Disc-Digital Audio (CD-DA) format stores audio information as a series of frames. Each frame contains a number of digital samples representing, for example, music, and a header that contains file format information. As shown in FIG. 1, according to an embodiment of the present invention some of the header information can be identified and "scrambled" using the predetermined key at steps 110 to 130. The music samples can remain unchanged. Using this technique, a traditional CD-DA player will be able to play a distorted version of the music in the sample. The amount of distortion will depend on the way, and extent, that the header, or file format, information has been scrambled. It would also be possible to instead scramble some of the digital samples while leaving the header information alone. In general, the digital signal would be protected by manipulating data at the inherent granularity, or "frames," of the CD-DA file format. To decode the information, a predetermined key is used before playing the digital information at steps 140 and 150.

A key-based decoder can act as a "plug-in" digital player of broadcast signal streams without foreknowledge of the encoded media stream. Moreover, the data format orientation is used to partially scramble data in transit to prevent unauthorized descrambled access by decoders that lack authorized keys. A distributed key can be used to unscramble the scrambled content because a decoder would understand how to process the key. Similar to on-the-fly decryption operations, the benefits inherent in this embodiment include the fact that the combination of watermarked content security, which is key-based, and the descrambling of the data, can be performed by the same key which can be a plurality of mask sets. The mask sets may include primary, convolution and message delimiter masks with file format data included.

The creation of an optimized "envelope" for insertion of watermarks provides the basis of much watermark security, but is also a complementary goal of the present invention. The predetermined or random key that is generated is not only an essential map to access the hidden information signal, but is also the descrambler of the previously scrambled signal's format for playback or viewing.

In a system requiring keys for watermarking content and validating the distribution of the content, different keys may be used to encode different information while secure one way hash functions or one-time pads may be incorporated to secure the embedded signal. The same keys can be used to later validate the embedded digital signature, or even fully decode the digital watermark if desired. Publishers can easily stipulate that content not only be digitally watermarked but that distributors must check the validity of the watermarks by performing digital signature checks with keys that lack any other functionality. The system can extend to simple authentication of text in other embodiments.

Before such a market is economically feasible, there are other methods for deploying key-based watermarking coupled with transfer functions to partially scramble the content to be distributed without performing full public key encryption, i.e., a key pair is not necessarily generated, simply, a predetermined key's function is created to re-map the data of the content file in a lossless process. Moreover, the scrambling performed by the present invention may be more dependent on the file in question. Dissimilarly, encryption is not specific to any particular media but is performed on data. The file format remains unchanged, rendering the file useable by any conventional viewer/player, but the signal quality can be intentionally degraded in the absence of the proper player and key. Public-key encryption seeks to completely obscure the sensitive "plaintext" to prevent comparisons with the "ciphertext" to determine a user's private keys. Centralized encryption only differs in the utilization of a single key for both encryption and decryption making the key even more highly vulnerable to attacks to defeat the encryption process. With the present invention, a highly sought after photograph may be hazy to the viewer using any number of commonly available, nonproprietary software or hardware, without the authorized key. Similarly, a commercially valuable song may sound poor.

The benefit of some form of cryptography is not lost in the present invention. In fact, some piracy can be deterred when the target signal may be known but is clearly being protected through scrambling. What is not anticipated by known techniques, is an ala carte method to change various aspects of file formatting to enable various "scrambled states" for content to be subsequently distributed. An image may lack all red pixels or may not have any of the most significant bits activated. An audio sample can similarly be scrambled to render it less-than-commercially viable.

The present invention also provides improvements over known network-based methods, such as those used for the streaming of media data over the Internet. By manipulating file formats, the broadcast media, which has been altered to "fit" within electronic distribution parameters, such as bandwidth availability and error correction considerations, can be more effectively utilized to restrict the subsequent use of the content while in transit as well as real-time viewing or playing.

The mask set providing the transfer function can be read on a per-use basis by issuing an authorized or authenticating "key" for descrambling the signal that is apparent to a viewer or a player or possessor of the authenticating key. The mask set can be read on a per-computer basis by issuing the authorized key that is more generalized for the computer that receives the broadcast signals. Metering and subscription models become viable advantages over known digital watermark systems which assist in designating the ownership of a copy of digitized media content, but do not prevent or restrict the copying or manipulation of the sampled signal in question. For broadcast or streamed media, this is especially the case. Message authentication is also possible, though not guaranteeing the same security as an encrypted file as with general crypto systems.

The present invention thus benefits from the proprietary player model without relying on proprietary players. No new players will be necessary and existing multimedia file formats can be altered to exact a measure of security which is further increased when coupled with digital watermarks. As with most consumer markets for media content, predominant file formats exist, de facto, and corresponding formats for computers likewise exist. For a commercial compact disc quality audio recording, or 16 bit 44.1 kHz, corresponding file formats include: Audio Interchange File Format (AIFF), Microsoft WAV, Sound Designer II, Sun's .au, Apple's Quicktime, etc. For still image media, formats are similarly abundant: TIFF, PICT, JPEG, GIF, etc. Requiring the use of additional proprietary players, and their complementary file formats, for limited benefits in security is wasteful. Moreover, almost all computers today are multimedia-capable, and this is increasingly so with the popularity of Intel's MMX chip architecture and the PowerPC line of microchips. Because file formatting is fundamental in the playback of the underlying data, the predetermined key can act both as a map, for information to be encoded as watermark data regarding ownership, and a descrambler of the file that has been distributed. Limitations will only exist in how large the key must be retrofitted for a given application, but any manipulation of file format information is not likely to exceed the size of data required versus that for an entire proprietary player.

As with previous disclosures by the inventor on digital watermarking techniques, the present invention may be implemented with a variety of cryptographic protocols to increase both confidence and security in the underlying system. A predetermined key is described as a set of masks. These masks may include primary, convolution and message delimiter mask. In previous disclosures, the functionality of these masks is defined solely for mapping. The present invention includes a mask set which is also controlled by the distributing party of a copy of a given media signal. This mask set is a transfer function which is limited only by the parameters of the file format in question. To increase the uniqueness or security of each key used to scramble a given media file copy, a secure one way hash function can be used subsequent to transfer properties that are initiated to prevent the forging of a particular key. Public and private keys may be used as key pairs to further increase the unlikeliness that a key may be compromised.

These same cryptographic protocols can be combined with the embodiments of the present invention in administering streamed content that requires authorized keys to correctly display or play the streamed content in an unscrambled manner. As with digital watermarking, symmetric or asymmetric public key pairs may be used in a variety of implementations. Additionally, the need for certification authorities to maintain authentic key-pairs becomes a consideration for greater security beyond symmetric key implementations. The cryptographic protocols makes possible, as well, a message of text to be authenticated by a message authenticating function in a general computing device that is able to ensure secure message exchanges between authorizing parties.

Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.

What is claimed is:

1. A method for copy protection of digital information, the digital information including a digital sample and format information, comprising the steps of:

identifying a portion of the format information to be encoded;

generating encoded format information from the identified portion of the format information; and

generating encoded digital information, including the digital sample and the encoded format information.

2. The method of claim 1, further comprising the step of requiring a predetermined key to decode the encoded format information.

3. The method of claim 2, wherein the digital sample and format information are configured to be used with a digital player, and wherein information output from the digital player will have a degraded quality unless the encoded format information is decoded with the predetermined key.

4. The method of claim 3, wherein the information output from the digital player represents a still image, audio or video.

5. The method of claim 3, wherein the information output represents text data to be authenticated.

6. A method for protecting a digital signal, the digital signal including digital samples in a file format having an inherent granularity, comprising the step of:

creating a predetermined key comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

7. The method of claim 6, wherein the digital signal represents a continuous analog waveform.

8. The method of claim 6, wherein the predetermined key comprises a plurality of mask sets.

9. The method of claim 6, wherein the digital signal is a message to be authenticated.

10. The method of claim 6, wherein the mask set is ciphered by a key pair comprising a public key and a private key.

11. The method of claim 6, further comprising the step of:

using a digital watermarking technique to encode information that identifies ownership, use, or other information about the digital signal, into the digital signal.

12. The method of claim 6, wherein the digital signal represents a still image, audio or video.

13. The method of claim 6, further comprising the steps of:

selecting the mask set, including one or more masks having random or pseudo-random series of bits; and

validating the mask set at the start of the transfer function-based mask set.

14. The method of claim 13, wherein said step of validating comprises the step of:

comparing a hash value computed at the start of the transfer function-based mask set with a determined transfer function of the hash value.

15. The method of claim 6, further comprising the steps of:

selecting the mask set, including one or more masks having random or pseudo-random series of bits; and

authenticating the mask set by comparing a hash value computed at the start of the transfer function-based mask set with a determined transfer function of the hash value.

16. The method of claim 13, wherein said step of validating comprises the step of:

comparing a digital signature at the start of the transfer function-based mask set with a determined transfer function of the digital signature.

17. The method of claim 6, further comprising the steps of:

selecting the mask set, including one or more masks having random or pseudo-random series of bits; and

authenticating the mask set by comparing a digital signature at the start of the transfer function-based mask set with a determined transfer function of the digital signature.

18. The method of claim 13, further comprising the step of:

using a digital watermarking technique to embed information that identifies ownership, use, or other information about the digital signal, into the digital signal; and

wherein said step of validating is dependent on validation of the embedded information.

19. The method of claim 6, further comprising the step of:

computing a secure one way hash function of carrier signal data in the digital signal, wherein the hash function is insensitive to changes introduced into the carrier signal for the purpose of carrying the transfer function-based mask set.

20. A method for protecting a digital signal, the digital signal including digital samples in a file format having an inherent granularity, comprising the steps of:

creating a predetermined key comprised of a transfer function-based mask set that can manipulate data at the inherent granularity of the file format of the underlying digitized samples;

authenticating the predetermined key containing the correct transfer function-based mask set during playback of the data; and

metering the playback of the data to monitor content.

21. The method of claim 20, wherein the predetermined key is authenticated to authenticate message information

22. A method to prepare for the scrambling of a sample stream of data, comprising the steps of:

generating a plurality of mask sets to be used for encoding, including a random primary mask, a random convolution mask and a random start of message delimiter;

obtaining a transfer function to be implemented;

generating a message bit stream to be encoded;

loading the message bit stream, a stega-cipher map truth table, the primary mask, the convolution mask and the start of message delimiter into memory;

initializing the state of a primary mask index, a convolution mask index, and a message bit index; and

setting a message size equal to the total number of bits in the message bit stream.

23. A method to prepare for the encoding of stega-cipher information into a sample stream of data, comprising the steps of:

generating a mask set to be used for encoding, the set including a random primary mask, a random convolution mask, and a random start of message delimiter;

obtaining a message to be encoded;

compressing and encrypting the message if desired;

generating a message bit stream to be encoded;

loading the message bit stream, a stega-cipher map truth table, the primary mask, the convolution mask and the start of message delimiter into memory;

initializing the state of a primary mask index, a convolution mask index, and a message bit index; and

setting the message size equal to the total number of bits in the message bit stream.

24. The method of claim 23 wherein the sample stream of data has a plurality of windows, further comprising the steps of:

calculating over which windows in the sample stream the message will be encoded;

computing a secure one way hash function of the information in the calculated windows, the hash function generating hash values insensitive to changes in the samples induced by a stega-cipher; and

encoding the computed hash values in an encoded stream of data.

25. The method of claim 13, wherein said step of selecting comprises the steps of:

collecting a series of random bits derived from keyboard latency intervals in random typing;

processing the initial series of random bits through an MD5 algorithm;

using the results of the MD5 processing to seed a triple-DES encryption loop;

cycling through the triple-DES encryption loop, extracting the least significant bit of each result after each cycle; and

concatenating the triple-DES output bits into the random series of bits.

26. A method for copy protection of digital information, the digital information including a digital sample and format information, comprising the steps of:

identifying a portion of the digital sample to be encoded;

generating an encoded digital sample from the identified portion of the digital sample; and

generating encoded digital information, including the encoded digital sample and the format information.

27. The method of claim 26, further comprising the step of requiring a predetermined key to decode the encoded digital sample.

28. The method of claim 27, wherein the digital sample and format information are configured to be used with a digital player, and wherein information output from the digital player will have a degraded quality unless the encoded digital sample is decoded with the predetermined key.

29. The method of claim 27, wherein information output will have non authentic message data unless the encode digital sample is decoded with the predetermined key.

ABSTRACT OF THE DISCLOSURE

A method for combining transfer functions with predetermined key creation. In one embodiment, digital information, including a digital sample and format information, is protected by identifying and encoding a portion of the format information. Encoded digital information, including the digital sample and the encoded format information, is generated to protect the original digital information. In another embodiment, a digital signal, including digital samples in a file format having an inherent granularity, is protected by creating a predetermined key. The predetermined key is comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

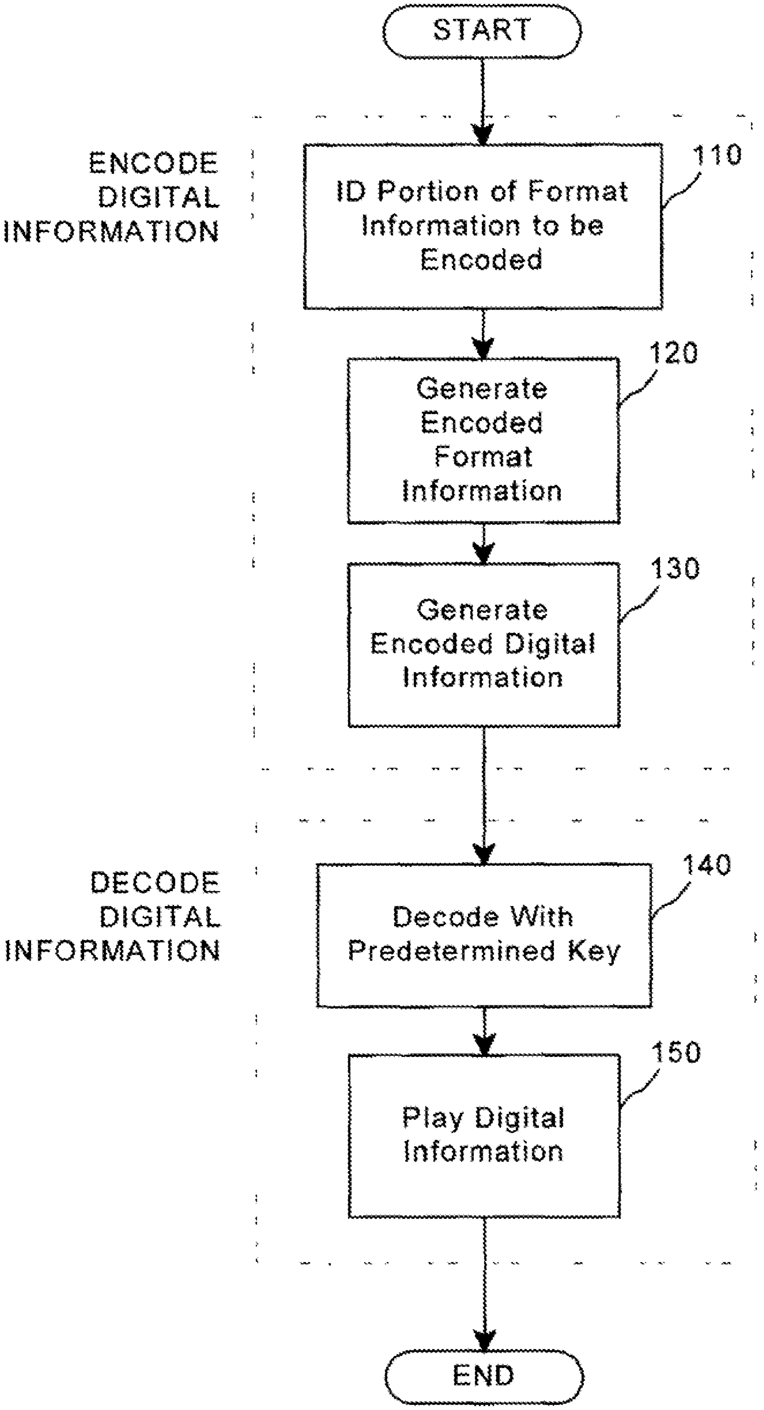

START
ENCODE DIGITAL INFORMATION
ID Portion of Format Information to be Encoded
110
Generate Encoded Format Information
120
Generate Encoded Digital Information
130
DECODE DIGITAL INFORMATION
Decode With Predetermined Key
140
Play Digital Information
150
END

FIG. 1

ATTORNEY'S DOCKET NO.
1607/6

DECLARATION AND POWER OF ATTORNEY - ORIGINAL APPLICATION

As below named inventors, we hereby declare that

Our residence, post office address, and citizenship are as stated below next to our name,

We believe we are the original, first, and joint inventors of the subject matter that is claimed and for which a patent is sought on the invention entitled **Method for Combining Transfer Functions with Predetermined Key Creation** filed herewith

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims.

We acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, § 1 56(a), a copy of which is attached

PRIOR UNITED STATES APPLICATION(S)

We hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application

APPLICATION NUMBER	FILING DATE (day, month, year)	STATUS (i e. Patented, Pending, Abandoned)
08/587,943	**17 January 1996**	**Pending**

POWER OF ATTORNEY: As named inventors, we hereby appoint the following attorneys John C Altmiller (Reg No 25,951); Frank V Pietrantonio (Reg No 37,966), and Patrick J Buckley (Reg No 40,928) of KENYON & KENYON with offices located at 1025 Connecticut Ave , N W , Washington, D C 20036, telephone (202) 429-1776, as my attorneys and/or agents with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith

SEND CORRESPONDENCE, AND DIRECT TELEPHONE CALLS TO

John C. Altmiller
KENYON & KENYON
1025 Connecticut Avenue, N.W.
Washington, D.C. 20036
(202) 429-1776 (phone)
(202) 429-0796 (facsimile)

I declare that all statements made herein of my own knowledge are true and all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under § 1001 of Title 18 of the United States Code and that such willful statements may jeopardize the validity of the application or any patent issuing thereon

FULL NAME OF INVENTOR	FAMILY NAME **MOSKOWITZ**	FIRST GIVEN NAME **SCOTT**	SECOND GIVEN NAME **A.**
RESIDENCE & CITIZENSHIP	CITY **MIAMI**	STATE OR FOREIGN COUNTRY **FLORIDA**	COUNTRY OF CITIZENSHIP **USA**
POST OFFICE ADDRESS	POST OFFICE ADDRESS **16711 COLLINS AVENUE #2505**	CITY **MIAMI**	STATE & ZIP CODE/COUNTRY **FL 33160**
Signature Scott Moskowitz		Date March 20, 1998	

154490

154490

Attachment 13 Page 29 of 31

§1.56 Duty to disclose information material to patentability.

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is cancelled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is cancelled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) Prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) The closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made of record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application;

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.

[57 FR 2034, Jan 17, 1992]

Applicant or Patentee: **Scott A. Moskowitz** **Attorney's Docket No.: 1607/6**
Serial or Patent No.: not yet assigned
Filed or Issued: herewith

Title: **METHOD FOR COMBINING TRANSFER FUNCTIONS WITH PREDETERMINED KEY CREATION**

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS (37 C.F.R. §§ 1.9(c) & 1.27(b)) – INDEPENDENT INVENTOR

As a below named inventor, I hereby declare that I qualify as an independent inventor as defined in 37 C.F.R. 1.9(c) for purposes of paying reduced fees under Section 41(a) and (b) of Title 35, United States Code, to the Patent and Trademark Office with regard to the invention entitled **METHOD FOR COMBINING TRANSFER FUNCTIONS WITH PREDETERMINED KEY CREATION** described in

☒ the specification filed herewith
□ application serial no. ___________, filed ______________.
□ patent no. ___________, issued ______________.

I have not assigned, granted, conveyed or licensed and am under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who could not be classified as an independent inventor under 37 C.F.R. 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 C.F.R. 1.9(d) or a nonprofit organization under 37 C.F.R. 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

☒ No such person, concern, or organization exists.
□ Each such person, concern or organization is listed below*

NOTE: *Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities (37 C.F.R. 1.27).*

FULL NAME: Scott A. Moskowitz
ADDRESS: 16711 Collins Avenue #2505, Miami, Florida 33160
☒ INDIVIDUAL □ SMALL BUSINESS CONCERN □ NONPROFIT ORGANIZATION

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 C.F.R. § 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Scott A. Moskowitz Date: March 20, 1998

154696

PTO/SB/05 (03-01)
Approved for use through 10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Please type a plus sign (+) inside this box → □

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	80391.0003/Con
First Inventor	MOSKOWITZ
Title	Method for Combining Transfer Functions...
Express Mail Label No.	

U.S. PTO
10/602777
06/25/03

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. ☑ Fee Transmittal Form (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)
2. ☑ Applicant claims small entity status.
See 37 CFR 1.27.
3. ☑ Specification [Total Pages 26]
(preferred arrangement set forth below)
   - Descriptive title of the invention
   - Cross Reference to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to sequence listing, a table, or a computer program listing appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings (if filed)
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
4. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 1]
5. Oath or Declaration [Total Pages]
   a. ☑ Newly executed (original or copy)
   b. □ Copy from a prior application (37 CFR 1.63 (d))
   (for continuation/divisional with Box 18 completed)
      i. □ DELETION OF INVENTOR(S)
      Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. ☑ Application Data Sheet. See 37 CFR 1.76
7. □ CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)
8. Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)
   a. □ Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
      i. □ CD-ROM or CD-R (2 copies); or
      ii. □ paper
   c. □ Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

9. □ Assignment Papers (cover sheet & document(s))
10. □ 37 CFR 3.73(b) Statement (when there is an assignee) □ Power of Attorney
11. □ English Translation Document (if applicable)
12. □ Information Disclosure Statement (IDS)/PTO-1449 □ Copies of IDS Citations
13. ☑ Preliminary Amendment
14. ☑ Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)
15. □ Certified Copy of Priority Document(s)
(if foreign priority is claimed)
16. □ Nonpublication Request under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. □ Other: ....

18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:

☑ Continuation □ Divisional □ Continuation-in-part (CIP) of prior application No.: 09 / 046,627

Prior application information: Examiner D. Meislahn Group / Art Unit: 2132

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. CORRESPONDENCE ADDRESS

☑ Customer Number or Bar Code Label 29693 or □ Correspondence address below

Name	PATENT TRADEMARK OFFICE				
Address					
City		State		Zip Code	
Country		Telephone		Fax	

Name (Print/Type)	Floyd B. Chapman, Esq.	Registration No. (Attorney/Agent)	40,555
Signature	Floyd B Chapman	Date	June 25, 2003

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

PTO/SB/17 (05-03)
Approved for use through 04/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2003

Effective 01/01/2003. Patent fees are subject to annual revision.

☑ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($) 468.00

Complete if Known

Application Number	Unassigned
Filing Date	June 24, 2003
First Named Inventor	MOSKOWITZ
Examiner Name	Unassigned
Art Unit	Unassigned
Attorney Docket No.	80391.0003/CON

METHOD OF PAYMENT *(check all that apply)*

□ Check □ Credit card □ Money Order □ Other □ None

☑ Deposit Account:

Deposit Account Number	50-1129
Deposit Account Name	Wiley Rein & Fielding LLP

The Director is authorized to: *(check all that apply)*

□ Charge fee(s) indicated below ☑ Credit any overpayments

☑ Charge any additional fee(s) during the pendency of this application

□ Charge fee(s) indicated below, **except for the filing fee**

to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	750	2001	375	Utility filing fee	375
1002	330	2002	165	Design filing fee	
1003	520	2003	260	Plant filing fee	
1004	750	2004	375	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($) 375.00

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

			Extra Claims		Fee from below		Fee Paid
Total Claims	21	-20** =	1	X	9	=	9
Independent Claims	5	- 3** =	2	X	42	=	84
Multiple Dependent						=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	84	2201	42	Independent claims in excess of 3
1203	280	2203	140	Multiple dependent claim, if not paid
1204	84	2204	42	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 93.00

***or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	410	2252	205	Extension for reply within second month	
1253	930	2253	465	Extension for reply within third month	
1254	1,450	2254	725	Extension for reply within fourth month	
1255	1,970	2255	985	Extension for reply within fifth month	
1401	320	2401	160	Notice of Appeal	
1402	320	2402	160	Filing a brief in support of an appeal	
1403	280	2403	140	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,300	2453	650	Petition to revive - unintentional	
1501	1,300	2501	650	Utility issue fee (or reissue)	
1502	470	2502	235	Design issue fee	
1503	630	2503	315	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	750	2809	375	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	750	2810	375	For each additional invention to be examined (37 CFR 1.129(b))	
1801	750	2801	375	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) ____________________

*Reduced by Basic Filing Fee Paid **SUBTOTAL (3)** ($)

SUBMITTED BY (Complete *(if applicable)*)

Name (Print/Type)	Floyd B. Chapman	*Registration No. (Attorney/Agent)*	40,555	*Telephone*	202.719.7000
Signature	Floyd B Chapman			*Date*	June 25, 2003

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

[illegible] assistance [illegible] 1-8[illegible]TO-9199 and select option 2.

wrf Wiley Rein & Fielding LLP

1776 K STREET NW
WASHINGTON, DC 20006
PHONE 202.719.7000
FAX 202.719.7049

Virginia Office
7925 JONES BRANCH DRIVE
SUITE 6200
McLEAN, VA 22102
PHONE 703.905.2800
FAX 703.905.2820

www.wrf.com

June 25, 2003

Floyd Chapman
202.719.7308
fchapman@wrf.com

VIA HAND DELIVERY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Re: New Continuation Application
(Continuation of 09/046,627)
Inventor: Scott Moskowitz
Title: **Method for Combining Transfer Functions with Predetermined Key Creation**
Attorney Docket: 80391.0003

Dear Sir:

Please accord the enclosed application a filing date and serial number.

Applicant hereby claims priority as U.S. Application Serial No. 09/046,627, filed in the U.S. Patent Office on 24 April 1998, the entire contents of which is hereby incorporated by reference into this new continuation application.

The following are attached:

1) Utility Application Transmittal Form (1 page);
2) Fee Transmittal Sheet authorizing a charge to our Deposit Account of $468.00 (1 page plus duplicate);
3) Application Data Sheet (1 page);
4) Preliminary Amendment (10 pages);
5) Original specification (27 pages total—17 pages specification; 8 pages claims; 1 page abstract; 1 sheets of drawings);
5) Declaration (3 pages);
6) Stamped return receipt postcard.

The undersigned authorizes the Commissioner to charge any additional fees to Deposit Account No. 50-1129.

Respectfully submitted,

Floyd B Chapman

Floyd Chapman, Esq.
Reg. No. 40,555

Please type a plus sign (+) inside this box → □

PTO/SB/05 (03-01)
Approved for use through 10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	80391.0003/Con
First Inventor	MOSKOWITZ
Title	Method for Combining Transfer Functions...
Express Mail Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. ☑ Fee Transmittal Form (e.g., PTO/SB/17) *(Submit an original and a duplicate for fee processing)*
2. ☑ Applicant claims small entity status. See 37 CFR 1.27.
3. ☑ Specification [*Total Pages* 26] *(preferred arrangement set forth below)*
   - Descriptive title of the Invention
   - Cross Reference to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to sequence listing, a table, or a computer program listing appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings *(if filed)*
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
4. ☒ Drawing(s) *(35 U.S.C. 113)* [*Total Sheets* 1]
5. Oath or Declaration [*Total Pages*]
   a. ☑ Newly executed (original or copy)
   b. □ Copy from a prior application (37 CFR 1.63 (d)) *(for continuation/divisional with Box 18 completed)*
      i. □ DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. ☑ Application Data Sheet. See 37 CFR 1.76
7. □ CD-ROM or CD-R in duplicate, large table or Computer Program *(Appendix)*
8. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. □ Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
      i. □ CD-ROM or CD-R (2 copies); or
      ii. □ paper
   c. □ Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

9. □ Assignment Papers (cover sheet & document(s))
10. □ 37 CFR 3.73(b) Statement *(when there is an assignee)* □ Power of Attorney
11. □ English Translation Document *(if applicable)*
12. □ Information Disclosure Statement (IDS)/PTO-1449 □ Copies of IDS Citations
13. ☑ Preliminary Amendment
14. ☑ Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
15. □ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. □ Nonpublication Request under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. □ Other: ....

18. If a CONTINUING APPLICATION, *check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:*

☑ Continuation □ Divisional □ Continuation-in-part (CIP) of prior application No.: 09 / 046,627

Prior application information: Examiner D. Meislahn Group / Art Unit: 2132

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. CORRESPONDENCE ADDRESS

☑ Customer Number or Bar Code Label 29693 PATENT TRADEMARK OFFICE or □ Correspondence address below

Name					
Address					
City		*State*		*Zip Code*	
Country		*Telephone*		*Fax*	

Name (Print/Type)	Floyd B. Chapman, Esq.	*Registration No. (Attorney/Agent)*	40,555
Signature	Floyd B. Chapman	*Date*	June 25, 2003

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

Attaachment 14 Page 5 of 51

PTO/SB/17 (05-03)
Approved for use through 04/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2003

Effective 01/01/2003. Patent fees are subject to annual revision.

☑ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($) 468.00

Complete if Known

Application Number	Unassigned
Filing Date	June 24, 2003
First Named Inventor	MOSKOWITZ
Examiner Name	Unassigned
Art Unit	Unassigned
Attorney Docket No.	80391.0003/CON

METHOD OF PAYMENT *(check all that apply)*

□ Check □ Credit card □ Money Order □ Other □ None

☑ Deposit Account:

Deposit Account Number	50-1129
Deposit Account Name	Wiley Rein & Fielding LLP

The Director is authorized to: *(check all that apply)*

□ Charge fee(s) indicated below ☑ Credit any overpayments

☑ Charge any additional fee(s) during the pendency of this application

□ Charge fee(s) indicated below, **except for the filing fee**

to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	750	2001	375	Utility filing fee	375
1002	330	2002	165	Design filing fee	
1003	520	2003	260	Plant filing fee	
1004	750	2004	375	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($) 375.00

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

			Extra Claims		Fee from below		Fee Paid
Total Claims	21	-20** =	1	X	9	=	9
Independent Claims	5	- 3** =	2	X	42	=	84
Multiple Dependent						=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	84	2201	42	Independent claims in excess of 3
1203	280	2203	140	Multiple dependent claim, if not paid
1204	84	2204	42	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 93.00

***or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	410	2252	205	Extension for reply within second month	
1253	930	2253	465	Extension for reply within third month	
1254	1,450	2254	725	Extension for reply within fourth month	
1255	1,970	2255	985	Extension for reply within fifth month	
1401	320	2401	160	Notice of Appeal	
1402	320	2402	160	Filing a brief in support of an appeal	
1403	280	2403	140	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,300	2453	650	Petition to revive - unintentional	
1501	1,300	2501	650	Utility issue fee (or reissue)	
1502	470	2502	235	Design issue fee	
1503	630	2503	315	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	750	2809	375	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	750	2810	375	For each additional invention to be examined (37 CFR 1.129(b))	
1801	750	2801	375	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) ____________

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($)

SUBMITTED BY (Complete *(if applicable)*)

Name (Print/Type)	Floyd B. Chapman	*Registration No. (Attorney/Agent)*	40,555	*Telephone*	202.719.7000
Signature	Floyd B Chapman			*Date*	June 25, 2003

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

wrf Wiley Rein & Fielding LLP

1776 K STREET NW
WASHINGTON, DC 20006
PHONE 202.719.7000
FAX 202.719.7049

Virginia Office
7925 JONES BRANCH DRIVE
SUITE 6200
McLEAN, VA 22102
PHONE 703.905.2800
FAX 703.905.2820

www.wrf.com

June 25, 2003

Floyd Chapman
202.719.7308
fchapman@wrf.com

VIA HAND DELIVERY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Re: New Continuation Application
(Continuation of 09/046,627)
Inventor: Scott Moskowitz
Title: **Method for Combining Transfer Functions with Predetermined Key Creation**
Attorney Docket: 80391.0003

Dear Sir:

Please accord the enclosed application a filing date and serial number.

Applicant hereby claims priority as U.S. Application Serial No. 09/046,627, filed in the U.S. Patent Office on 24 April 1998, the entire contents of which is hereby incorporated by reference into this new continuation application.

The following are attached:

1) Utility Application Transmittal Form (1 page);
2) Fee Transmittal Sheet authorizing a charge to our Deposit Account of $468.00 (1 page plus duplicate);
3) Application Data Sheet (1 page);
4) Preliminary Amendment (10 pages);
5) Original specification (27 pages total—17 pages specification; 8 pages claims; 1 page abstract; 1 sheets of drawings);
5) Declaration (3 pages);
6) Stamped return receipt postcard.

The undersigned authorizes the Commissioner to charge any additional fees to Deposit Account No. 50-1129.

Respectfully submitted,

Floyd B Chapman

Floyd Chapman, Esq.
Reg. No. 40,555

WRFMAIN 12091216.1

Application Data Sheet

Inventor Information

Inventor One Given Name: Scott A.
Family Name: MOSKOWITZ
Postal Address: 16711 Collins Avenue, #2505
City: Miami
State/Province: Florida
Postal or Zip Code: 33160
Citizenship: US

Correspondence Information

Correspondence Customer Number:	29693
Telephone Number One:	(202) 719-7000
Facsimile Number:	(202) 719-7049
E-Mail Address:	fchapman@wrf.com

Application Information

Title Line One: METHOD FOR COMBINING TRANSFER FUNCTIONS WITH PREDETERMINED KEY CREATION
Total Drawing Sheets: 1 sheet
Docket Number: 80391.0003
Application Type: Continuation Application
Formal Drawings: Yes

Representative Information

Representative Customer Number: 29693

Domestic Priority

This application is a: continuation application of U.S. Application Serial No. 09/046,627 filed 24 April 1998.

Prior Foreign Applications

Foreign Application One:
Filing Date:
Country:
Priority Claimed:
WRFMAIN 12091249.1

PATENT
Atty. Docket: 80391.0003/Con

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	
Scott MOSKOWITZ	)	Group Art Unit: Unassigned
	)	
Application Number: Unassigned	)	Examiner: Unassigned
	)	
Filed: Herewith	)	
	)	
Title: Method for Combining Transfer Functions with Predetermined Key Creation	)	

BOX Patent Application
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PRELIMINARY AMENDMENT

Sir:

Prior to examination on the merits, please enter the following amendments to the application.

IN THE SPECIFICATION:

Please delete the section entitled "CROSS-REFERENCE TO RELATED APPLICATIONS" on page 1, lines 10-14, of the originally filed application and insert the new section entitled "CROSS-REFERENCE TO RELATED APPLICATIONS" on page 1, at line 5:

--This application is a continuation application of U.S. Patent Application Serial No. 09/046,627 (now awaiting issuance), which is a continuation of U.S. Patent Application Serial No. 08/587,943, filed January 17, 1996, (which issued April 28, 1998, as U.S. Patent No. 5,745,943). The entire disclosure of U.S. Patent Application No. 09/046,627 is hereby incorporated by reference.--

IN THE CLAIMS:

Please cancel claims 1-5 and 26-29 without prejudice to Applicant's right to seek allowance of said claims in a related application.

Please amend claims as indicated below.

Claims 1-5 canceled without prejudice

6. (currently amended) A method for protecting a digital signal, comprising the steps of:

providing a ~~the~~ digital signal ~~including~~ comprising digital data and ~~samples in a~~ file format information; ~~having an inherent granularity, comprising the step of:~~

creating a predetermined key that manipulates the file format information~~comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples~~; and

manipulating the file format information using the predetermined key.

7. (original) The method of claim 6, wherein the digital signal represents a continuous analog waveform.

8. (original) The method of claim 6, wherein the predetermined key comprises a plurality of mask sets.

9. (original) The method of claim 6, wherein the digital signal is a message to be authenticated.

10. (currently amended) The method of claim 6, wherein the predetermined key comprises ~~mask set is ciphered by a~~ key pair comprising a public key and a private key.

11. (original) The method of claim 6, further comprising the step of:

using a digital watermarking technique to encode information that identifies ownership, use, or other information about the digital signal, into the digital signal.

12. (original) The method of claim 6, wherein the digital signal represents a still image, audio or video.

13. (currently amended) The method of claim 6, wherein the predetermined key comprises one or more mask sets having random or pseudo-random series of bits, the method further comprising the steps of:

~~selecting the mask set, including one or more masks having random or pseudo-random series of bits; and~~

validating the one or more mask sets ~~at the start of the transfer function-based mask set~~before manipulating the file format information using the predetermined key.

14. (currently amended) The method of claim 6, wherein the predetermined key comprises one or more mask sets having random or pseudo-random series of bits, the method further comprising the steps of:

~~selecting the mask set, including one or more masks having random or pseudo-random series of bits; and~~

validating the one or more mask sets ~~at the start of the transfer function-based mask set~~before manipulating the file format information using the predetermined key.

15. (currently amended) The method of claim 6, wherein the predetermined key comprises one or more mask sets having random or pseudo-random series of bits, the method further comprising the steps of:

~~selecting the mask set, including one or more masks having random or pseudo-random series of bits; and~~

generating a hash value using the one or more masks sets; and

authenticating the one or more mask sets by comparing the generated ~~a~~ hash value with a predetermined ~~computed at the start of the transfer function-based mask set with a determined transfer function of the~~ hash value.

16. (currently amended) The method of claim 13, wherein said step of validating comprises the steps of:

generating a digital signature using the one or more mask sets; and

comparing the digital signature with a predetermined digital signature.

~~comparing a digital signature at the start of the transfer function-based mask set with a determined transfer function of the digital signature.~~

17. (currently amended) The method of claim 6, wherein the predetermined key comprises one or more mask sets having random or pseudo-random series of bits, the method further comprising the ~~steps~~ step of:

~~selecting the mask set, including one or more masks having random or pseudo-random series of bits; and~~

authenticating the one or more mask sets by comparing a generated digital signature ~~at the start of the transfer function-based mask set~~ with a predetermined ~~transfer function of the~~ digital signature.

18. (original) The method of claim 13, further comprising the step of:

using a digital watermarking technique to embed information that identifies ownership, use, or other information about the digital signal, into the digital signal; and

wherein said step of validating is dependent on validation of the embedded information.

19. (currently amended) The method of claim 6, further comprising the step of:

computing a secure way hash function of carrier signal data in the digital signal, wherein the has function is insensitive to changes introduced into the carrier signal ~~for the purpose of carrying the transfer function-based mask set~~ during file format manipulation.

20. (currently amended) A method for protecting a digital signal, ~~the digital signal including digital samples in a file format having an inherent granularity,~~ comprising the steps of:

providing a digital signal comprising digital data and file format information;

creating a predetermined key ~~comprised~~ comprising a mask set ~~of a transfer function-based mask set that can manipulate data at the inherent granularity of the file format of the underlying digitized samples~~;

manipulating the file format information using the predetermined key;

authenticating the predetermined key ~~containing the correct transfer function-based mask set~~ during playback of the digital data; and

metering the playback of the digital data to monitor content.

21. (currently amended) The method of claim 20, wherein the predetermined key is authenticated to authenticate message information.

22. (currently amended) A method to prepare for the scrambling of a sample stream of data, comprising the steps of:

generating a plurality of mask sets to be used for encoding, including a random primary mask, a random convolution mask and a random start of message delimiter;

obtaining file format information about the sample stream of data; ~~a transfer function to be implemented;~~

generating a message bit stream to be encoded;

loading the message bit stream, a stega-cipher map truth table, the primary mask, the convolution mask and the start of the message delimiter into memory;

initializing the state of a primary mask index, a convolution mask index, and a message bit index; and

setting a message size equal to the total number of bits in the message bit stream.

23. (original) A method to prepare for the encoding of stega-cipher information into a sample stream of data, comprising the steps of:

generating a mask set to be used for encoding, the set including a random primary mask, a random convolution mask, and a random start of message delimiter;

obtaining a message to be encoded;

compressing and encrypting the message if desired;

generating a message bit stream to be encoded;

loading the message bit stream, a stega-cipher map truth table, the primary mask, the convolution mask and the start of message delimiter into memory;

initializing the state of a primary mask index, a convolution mask index, and a message bit index; and

setting the message size equal to the total number of bits in the message bit stream.

24. (original) The method of claim 23 wherein the sample stream of data has a plurality of windows, further comprising the steps of:

calculating over which windows in the sample stream the message will be encoded;

computing a secure one way hash function of the information in the calculated windows, the hash function generating hash values insensitive to changes in the samples induced by a stega-cipher; and

encoding the computed hash values in an encoded stream of data.

Claims 25-29 (canceled without prejudice)

30. (new) A method for protecting digital data, where the digital data signal is organized into a plurality of frames, each frame having i) a header comprising file format information and ii) at least a portion of the digital data, said method comprising the steps of:

creating a predetermined key to manipulate the file format information in one or more of the plurality of frames; and

manipulating the file format information using the predetermined key in at least two of the plurality of frames, such that the digital data will be perceived by a human as noticeably altered if it is played without using a decode key to restore the file format information to a prior state.

31. (new) The method of claim 30, wherein the predetermined key comprises a private key that is associated with a key pair.

REMARKS

Applicant requests entry of the amendments and submits that this application is in condition for allowance, and a notice to this effect is earnestly sought.

If the Examiner believes that prosecution might be furthered by discussing the application with Applicant's representatives, in person or by telephone, we would welcome the opportunity to do so.

Respectfully submitted,

WILEY REIN & FIELDING

Date: June 25, 2003

By: Floyd B Chapman
Floyd B. Chapman, No. 40,555

Wiley Rein & Fielding
1776 K Street, N.W.
Washington, D.C. 20006
Telephone: **(202) 220-7000**
Facsimile: **(202) 220-7049**

WRFMAIN 12082033.4

METHOD FOR COMBINING TRANSFER FUNCTIONS WITH PREDETERMINED KEY CREATION

FIELD OF THE INVENTION

The invention relates to the protection of digital information. More particularly, the invention relates to a method for combining transfer functions with predetermined key creation.

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. patent application Serial No. 08/587,943, filed January 17, 1996, entitled "Method for Stega-Cipher Protection of Computer Code," the entire disclosure of which is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

Increasingly, commercially valuable information is being created and stored in "digital" form. For example, music, photographs and video can all be stored and transmitted as a series of numbers, such as 1's and 0's. Digital techniques let the original information be recreated in a very accurate manner. Unfortunately, digital techniques also let the information be easily copied without the information owner's permission.

Because unauthorized copying is clearly a disincentive to the digital distribution of valuable information, it is important to establish responsibility for copies and derivative copies of such works. For example, if each authorized digital copy of a popular song is identified with a unique number, any unauthorized copy of the song would also contain the number. This would allow the owner of the information, such as a song publisher, to investigate who made the unauthorized copy. Unfortunately, it is possible that the unique number could be erased or altered if it is simply tacked on at the beginning or end of the digital information.

As will be described, known digital "watermark" techniques give creators and publishers of digitized multimedia content localized, secured identification and authentication of that content. In considering the various forms of multimedia content, such as "master," stereo, National Television Standards Committee (NTSC) video, audio tape or compact disc, tolerance of quality will vary with individuals and affect the underlying commercial and aesthetic value of the content. For example, if a digital version of a popular song sounds distorted, it will be less valuable to users. It is therefore desirable to embed copyright, ownership or purchaser information, or some combination of these and related data, into the content in a way that will damage the content if the watermark is removed without authorization.

To achieve these goals, digital watermark systems insert ownership information in a way that causes little or no noticeable effects, or "artifacts," in the underlying content signal. For example, if a digital watermark is inserted into a digital version of a song, it is important that a listener not be bothered by the slight changes introduced by the watermark. It is also important for the watermark technique to maximize the encoding level and "location sensitivity" in the signal to force damage to the content signal when removal is attempted. Digital watermarks address many of these concerns, and research in the field has provided extremely robust and secure implementations.

What has been overlooked in many applications described in the art, however, are systems which closely mimic distribution of content as it occurs in the real world. For instance, many watermarking systems require the original un-watermarked content signal to enable detection or decode operations. These include highly publicized efforts by NEC, Digimarc and others. Such techniques are problematic because, in the real world, original master copies reside in a rights holders vaults and are not readily available to the public.

With much activity overly focused on watermark survivability, the security of a digital watermark is suspect. Any simple linear operation for encoding information into a signal may be used to erase the embedded signal by inverting the process. This is not a difficult task, especially when detection software is a plug-in freely available to the public, such as with Digimarc. In general, these systems seek to embed cryptographic information, not cryptographically embed information into target media content.

Other methods embed ownership information that is plainly visible in the media signal, such as the method described in US Patent No. 5,530,739 to Braudaway et al. The system described in Braudaway protects a digitized image by encoding a visible watermark to deter piracy. Such an implementation creates an immediate weakness in securing the embedded information because the watermark is plainly visible. Thus, no search for the embedded signal is necessary and the watermark can be more easily removed or altered. For example, while certainly useful to some rights owners, simply placing the symbol "©" in the digital information would only provide limited protection. Removal by adjusting the brightness of the pixels forming the "©" would not be difficult with respect to the computational resources required.

Other relevant prior art includes US Patents No. 4,979,210 and 5,073,925 to Nagata et al., which encodes information by modulating an audio signal in the amplitude/time domain. The modulations introduced in the Nagata process carry a "copy/don't copy" message, which is easily found and circumvented by one skilled in the art. The granularity of encoding is fixed by the amplitude and frequency modulation limits required to maintain inaudibility. These limits are relatively low, making it impractical to encode more information using the Nagata process.

Although US Patent No. ~~5,664,018 to Leighton describes a means~~ to prevent collusion attacks in digital watermarks, the disclosed method may not actually provide the security described. For example, in cases where the watermarking technique is linear, the "insertion envelope" or "watermarking space" is well-defined and thus susceptible to attacks less sophisticated than collusion by unauthorized parties. Over-encoding at the watermarking encoding level is but one simple attack in such linear implementations. Another consideration not made by Leighton is that commercially-valuable content may already exist in a un-watermarked form somewhere, easily accessible to potential pirates, gutting the need for any type of collusive activity. Digitally signing the embedded signal with preprocessing of watermark data is more likely to prevent successful collusion. Furthermore, a "baseline" watermark as disclosed is quite subjective. It is simply described elsewhere in the art as the "perceptually significant" regions of a signal. Making a watermarking function less linear or inverting the insertion of watermarks would seem to provide the same benefit without the additional work required to create a "baseline" watermark. Indeed, watermarking algorithms should already be capable of defining a target insertion envelope or region without additional steps. What is evident is the Leighton patent does not allow for initial prevention of attacks on an embedded watermark as the content is visibly or audibly unchanged.

It is also important that any method for providing security also function with broadcasting media over networks such as the Internet, which is also referred to as "streaming." Commercial "plug-in" products such as RealAudio and RealVideo, as well as applications by vendors VDONet and Xtreme, are common in such network environments. Most digital watermark implementations focus on common file base signals and fail to anticipate the security of streamed signals. It is desirable that any protection scheme be able to function with a plug-in player without advanced knowledge of the encoded media stream.

Other technologies focus solely on file-based security. These technologies illustrate the varying applications for security that must be evaluated for different media and distribution environments. Use of cryptolopes or cryptographic containers, as proposed by IBM in its Cryptolope product, and InterTrust, as described in U.S. Patents No. 4,827,508, 4,977,594, 5,050,213 and 5,410,598, may discourage certain forms of piracy. Cryptographic containers, however, require a user to subscribe to particular decryption software to decrypt data. IBM's InfoMarket and InterTrust's DigiBox, among other implementations, provide a generalized model and need proprietary architecture to function. Every user must have a subscription or registration with the party which encrypts the data. Again, as a form of general encryption, the data is scrambled or encrypted without regard to the media and its formatting. Finally, control over copyrights or other neighboring rights is left with the implementing party, in this case, IBM, InterTrust or a similar provider.

Methods similar to these "trusted systems" exist, and Cerberus Central Limited and Liquid Audio, among a number of companies, offer systems which may functionally be thought of as subsets of IBM and InterTrust's more generalized security offerings. Both Cerberus and Liquid Audio propose proprietary player software which is registered to the user and "locked" in a manner parallel to the locking of content that is distributed via a cryptographic container. The economic trade-off in this model is that users are required to use each respective companies' proprietary player to play or otherwise manipulate content that is downloaded. If, as is the case presently, most music or other media is not available via these proprietary players and more companies propose non-compatible player formats, the proliferation of players will continue. Cerberus and Liquid Audio also by way of extension of their architectures provide for "near-CD quality" but proprietary compression. This requirement stems from the necessity not to allow content that has near-identical data make-up to an existing consumer electronic standard, in Cerberus and Liquid Audio's case the so-called Red Book audio CD standard of 16 bit 44.1 kHz, so that comparisons with the proprietary file may not yield how the player is secured. Knowledge of the player's file format renders its security ineffective as a file may be replicated and played on any common player, not the intended proprietary player of the provider of previously secured and uniquely formatted content. This is the parallel weakness to public key crypto-systems which have gutted security if enough plain text and cipher text comparisons enable a pirate to determine the user's private key.

Many approaches to digital watermarking leave detection and decoding control with the implementing party of the digital watermark, not the creator of the work to be protected. A set of secure digital watermark implementations address this fundamental control issue forming the basis of key-based approaches. These are covered by the following patents and pending applications, the entire disclosures of which are hereby incorporated by reference: US Patent No. 5,613, 004 entitled "Steganographic Method and Device" and its derivative US patent application Serial No. 08/775,216, US patent application Serial No. 08/587,944 entitled "Human Assisted Random Key Generation and Application for Digital Watermark System," US Patent Application Serial No. 08/587,943 entitled "Method for Stega-Cipher Protection of Computer Code," US patent application Serial No. 08/677,435 entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data," and US Patent Application Serial No. 08/772,222 entitled "Z-Transform Implementation of Digital Watermarks." Public key crypto-systems are described in US Patents No. 4,200,770, 4,218,582, 4,405,829 and 4,424,414, the entire disclosures of which are also hereby incorporated by reference.

In particular, an improved protection scheme is described in "Method for Stega-Cipher Protection of Computer Code," US patent application Serial No. 08/587,943. This technique uses the key-based insertion of binary executable computer code within a content signal that is subsequently, and necessarily, used to play or otherwise manipulate the signal in which it is encoded. With this system, however, certain computational requirements, such as one digital player per digital copy of content, may be necessitated. For instance, a consumer may download many copies of watermarked content. With this technique, the user would also be downloading as many copies of the digital player program. While this form of security may be desirable for some applications, it is not appropriate in many circumstances.

Finally, even when digital information is distributed in encoded form, it may be desirable to allow unauthorized users to play the information with a digital player, perhaps with a reduced level of quality. For example, a popular song may be encoded and freely distributed in encoded form to the public. The public, perhaps using commonly available plug-in digital players, could play the encoded content and hear the music in some degraded form. The music may sound choppy, or fuzzy or be degraded in some other way. This lets the public decide, based on the available lower quality version of the song, if they want to purchase a key from the publisher to decode, or "clean-up," the content. Similar approaches could be used to distribute blurry pictures or low quality video. Or even "degraded" text, in the sense that only authenticated portions of the text can be determined with the predetermined key or a validated digital signature for the intended message.

In view of the foregoing, it can be appreciated that a substantial need exists for a method allowing encoded content to be played, with degraded quality, by a plug-in digital player, and solving the other problems discussed above.

SUMMARY OF THE INVENTION

The disadvantages of the art are alleviated to a great extent by a method for combining transfer functions with predetermined key creation. In one embodiment, digital information, including a digital sample and format information, is protected by identifying and encoding a portion of the format information. Encoded digital information, including the digital sample and the encoded format information, is generated to protect the original digital information.

In another embodiment, a digital signal, including digital samples in a file format having an inherent granularity, is protected by creating a predetermined key. The predetermined key is comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

With these and other advantages and features of the invention that will become hereinafter apparent, the nature of the invention may be more clearly understood by reference to the following detailed description of the invention, the appended claims and to the several drawings attached herein.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block flow diagram of a method for copy protection or authentication of digital information according to an embodiment of the present invention.

DETAILED DESCRIPTION

In accordance with an embodiment of the present invention, a method combines transfer functions with predetermined key creation. Increased security is achieved in the method by combining elements of "public-key steganography" with cryptographic protocols, which keep in-transit data secure by scrambling the data with "keys" in a manner that is not apparent to those with access to the content to be distributed. Because different forms of randomness are combined to offer robust, distributed security, the present invention addresses an architectural "gray space" between two important areas of security: digital watermarks, a subset of the more general art of steganography, and cryptography. One form of randomness exists in the mask sets that are randomly created to map watermark data into an otherwise unrelated digital signal. The second form of randomness is the random permutations of data formats used with digital players to manipulate the content with the predetermined keys. These forms can be thought of as the transfer function versus the mapping function inherent to digital watermarking processes.

According to an embodiment of the present invention, a predetermined, or randomly generated, key is used to scramble digital information in a way that is unlike known "digital watermark" techniques and public key crypto-systems. As used herein, a key is also referred to as a "mask set" which includes one or more random or pseudo-random series of bits. Prior to encoding, a mask can be generated by any cryptographically secure random generation process. A block cipher, such as a Data Encryption Standard (DES) algorithm, in combination with a sufficiently random seed value, such as one created using a Message Digest 5 (MD5) algorithm, emulates a cryptographically secure random bit generator. The keys are saved in a database, along with information matching them to the digital signal, for use in descrambling and subsequent viewing or playback. Additional file format or transfer property information is prepared and made available to the encoder, in a bit addressable manner. As well, any authenticating function can be combined, such as Digital Signature Standard (DSS) or Secure Hash Algorithm (SHA).

Using the predetermined key comprised of a transfer function-based mask set, the data representing the original content is manipulated at the inherent granularity of the file format of the underlying digitized samples. Instead of providing, or otherwise distributing, watermarked content that is not noticeably altered, a partially "scrambled" copy of the content is distributed. The key is necessary both to register the sought-after content and to descramble the content into its original form.

The present invention uses methods disclosed in "Method for Stega-Cipher Protection of Computer Code," US Patent Application Serial No. 08/587,943, with respect to transfer functions related to the common file formats, such as PICT, TIFF, AIFF, WAV, etc. Additionally, in cases where the content has not been altered beyond being encoded with such functional data, it is possible for a digital player to still play the content because the file format has not been altered. Thus, the encoded content could still be played by a plug-in digital player as discrete, digitally sampled signals, watermarked or not. That is, the structure of the file can remain basically unchanged by the watermarking process, letting common file format based players work with the "scrambled" content.

For example, the Compact Disc-Digital Audio (CD-DA) format stores audio information as a series of frames. Each frame contains a number of digital samples representing, for example, music, and a header that contains file format information. As shown in FIG. 1, according to an embodiment of the present invention some of the header information can be identified and "scrambled" using the predetermined key at steps 110 to 130. The music samples can remain unchanged. Using this technique, a traditional CD-DA player will be able to play a distorted version of the music in the sample. The amount of distortion will depend on the way, and extent, that the header, or file format, information has been scrambled. It would also be possible to instead scramble some of the digital samples while leaving the header information alone. In general, the digital signal would be protected by manipulating data at the inherent granularity, or "frames," of the CD-DA file format. To decode the information, a predetermined key is used before playing the digital information at steps 140 and 150.

A key-based decoder can act as a "plug-in" digital player of broadcast signal streams without foreknowledge of the encoded media stream. Moreover, the data format orientation is used to partially scramble data in transit to prevent unauthorized descrambled access by decoders that lack authorized keys. A distributed key can be used to unscramble the scrambled content because a decoder would understand how to process the key. Similar to on-the-fly decryption operations, the benefits inherent in this embodiment include the fact that the combination of watermarked content security, which is key-based, and the descrambling of the data, can be performed by the same key which can be a plurality of mask sets. The mask sets may include primary, convolution and message delimiter masks with file format data included.

The creation of an optimized "envelope" for insertion of watermarks provides the basis of much watermark security, but is also a complementary goal of the present invention. The predetermined or random key that is generated is not only an essential map to access the hidden information signal, but is also the descrambler of the previously scrambled signal's format for playback or viewing.

In a system requiring keys for watermarking content and validating the distribution of the content, different keys may be used to encode different information while secure one way hash functions or one-time pads may be incorporated to secure the embedded signal. The same keys can be used to later validate the embedded digital signature, or even fully decode the digital watermark if desired. Publishers can easily stipulate that content not only be digitally watermarked but that distributors must check the validity of the watermarks by performing digital signature-checks with keys that lack any other functionality. The system can extend to simple authentication of text in other embodiments.

Before such a market is economically feasible, there are other methods for deploying key-based watermarking coupled with transfer functions to partially scramble the content to be distributed without performing full public key encryption, i.e., a key pair is not necessarily generated, simply, a predetermined key's function is created to re-map the data of the content file in a lossless process. Moreover, the scrambling performed by the present invention may be more dependent on the file in question. Dissimilarly, encryption is not specific to any particular media but is performed on data. The file format remains unchanged, rendering the file useable by any conventional viewer/player, but the signal quality can be intentionally degraded in the absence of the proper player and key. Public-key encryption seeks to completely obscure the sensitive "plaintext" to prevent comparisons with the "ciphertext" to determine a user's private keys. Centralized encryption only differs in the utilization of a single key for both encryption and decryption making the key even more highly vulnerable to attacks to defeat the encryption process. With the present invention, a highly sought after photograph may be hazy to the viewer using any number of commonly available, nonproprietary software or hardware, without the authorized key. Similarly, a commercially valuable song may sound poor.

The benefit of some form of cryptography is not lost in the present invention. In fact, some piracy can be deterred when the target signal may be known but is clearly being protected through scrambling. What is not anticipated by known techniques, is an ala carte method to change various aspects of file formatting to enable various "scrambled states" for content to be subsequently distributed. An image may lack all red pixels or may not have any of the most significant bits activated. An audio sample can similarly be scrambled to render it less-than-commercially viable.

The present invention also provides improvements over known network-based methods, such as those used for the streaming of media data over the Internet. By manipulating file formats, the broadcast media, which has been altered to "fit" within electronic distribution parameters, such as bandwidth availability and error correction considerations, can be more effectively utilized to restrict the subsequent use of the content while in transit as well as real-time viewing or playing.

The mask set providing the transfer function can be read on a per-use basis by issuing an authorized or authenticating "key" for descrambling the signal that is apparent to a viewer or a player or possessor of the authenticating key. The mask set can be read on a per-computer basis by issuing the authorized key that is more generalized for the computer that receives the broadcast signals. Metering and subscription models become viable advantages over known digital watermark systems which assist in designating the ownership of a copy of digitized media content, but do not prevent or restrict the copying or manipulation of the sampled signal in question. For broadcast or streamed media, this is especially the case. Message authentication is also possible, though not guaranteeing the same security as an encrypted file as with general crypto systems.

The present invention thus benefits from the proprietary player model without relying on proprietary players. No new players will be necessary and existing multimedia file formats can be altered to exact a measure of security which is further increased when coupled with digital watermarks. As with most consumer markets for media content, predominant file formats exist, de facto, and corresponding formats for computers likewise exist. For a commercial compact disc quality audio recording, or 16 bit 44.1 kHz, corresponding file formats include: Audio Interchange File Format (AIFF), Microsoft WAV, Sound Designer II, Sun's .au, Apple's Quicktime, etc. For still image media, formats are similarly abundant: TIFF, PICT, JPEG, GIF, etc. Requiring the use of additional proprietary players, and their complementary file formats, for limited benefits in security is wasteful. Moreover, almost all computers today are multimedia-capable, and this is increasingly so with the popularity of Intel's MMX chip architecture and the PowerPC line of microchips. Because file formatting is fundamental in the playback of the underlying data, the predetermined key can act both as a map, for information to be encoded as watermark data regarding ownership, and a descrambler of the file that has been distributed. Limitations will only exist in how large the key must be retrofitted for a given application, but any manipulation of file format information is not likely to exceed the size of data required versus that for an entire proprietary player.

As with previous disclosures by the inventor on digital watermarking techniques, the present invention may be implemented with a variety of cryptographic protocols to increase both confidence and security in the underlying system. A predetermined key is described as a set of masks. These masks may include primary, convolution and message delimiter mask. In previous disclosures, the functionality of these masks is defined solely for mapping. The present invention includes a mask set which is also controlled by the distributing party of a copy of a given media signal. This mask set is a transfer function which is limited only by the parameters of the file format in question. To increase the uniqueness or security of each key used to scramble a given media file copy, a secure one way hash function can be used subsequent to transfer properties that are initiated to prevent the forging of a particular key. Public and private keys may be used as key pairs to further increase the unlikeliness that a key may be compromised.

These same cryptographic protocols can be combined with the embodiments of the present invention in administering streamed content that requires authorized keys to correctly display or play the streamed content in an unscrambled manner. As with digital watermarking, symmetric or asymmetric public key pairs may be used in a variety of implementations. Additionally, the need for certification authorities to maintain authentic key-pairs becomes a consideration for greater security beyond symmetric key implementations. The cryptographic protocols makes possible, as well, a message of text to be authenticated by a message authenticating function in a general computing device that is able to ensure secure message exchanges between authorizing parties.

Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.

What is claimed is:

1. A method for copy protection of digital information, the digital information including a digital sample and format information, comprising the steps of:

identifying a portion of the format information to be encoded;

generating encoded format information from the identified portion of the format information; and

generating encoded digital information, including the digital sample and the encoded format information.

2. The method of claim 1, further comprising the step of requiring a predetermined key to decode the encoded format information.

3. The method of claim 2, wherein the digital sample and format information are configured to be used with a digital player, and wherein information output from the digital player will have a degraded quality unless the encoded format information is decoded with the predetermined key.

4. The method of claim 3, wherein the information output from the digital player represents a still image, audio or video.

5. The method of claim 3, wherein the information output represents text data to be authenticated.

6. A method for protecting a digital signal, the digital signal including digital samples in a file format having an inherent granularity, comprising the step of:

creating a predetermined key comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

7. The method of claim 6, wherein the digital signal represents a continuous analog waveform.

8. The method of claim 6, wherein the predetermined key comprises a plurality of mask sets.

9. The method of claim 6, wherein the digital signal is a message to be authenticated.

10. The method of claim 6, wherein the mask set is ciphered by a key pair comprising a public key and a private key.

11. The method of claim 6, further comprising the step of:

using a digital watermarking technique to encode information that identifies ownership, use, or other information about the digital signal, into the digital signal.

12. The method of claim 6, wherein the digital signal represents a still image, audio or video.

13. The method of claim 6, further comprising the steps of:

selecting the mask set, including one or more masks having random or pseudo-random series of bits; and

validating the mask set at the start of the transfer function-based mask set.

14. The method of claim 13, wherein said step of validating comprises the step of:

comparing a hash value computed at the start of the transfer function-based mask set with a determined transfer function of the hash value.

15. The method of claim 6, further comprising the steps of:

selecting the mask set, including one or more masks having random or pseudo-random series of bits; and

authenticating the mask set by comparing a hash value computed at the start of the transfer function-based mask set with a determined transfer function of the hash value.

16. The method of claim 13, wherein said step of validating comprises the step of:

comparing a digital signature at the start of the transfer function-based mask set with a determined transfer function of the digital signature.

17. The method of claim 6, further comprising the steps of:

selecting the mask set, including one or more masks having random or pseudo-random series of bits; and

authenticating the mask set by comparing a digital signature at the start of the transfer function-based mask set with a determined transfer function of the digital signature.

18. The method of claim 13, further comprising the step of:

using a digital watermarking technique to embed information that identifies ownership, use, or other information about the digital signal, into the digital signal; and

wherein said step of validating is dependent on validation of the embedded information.

19. The method of claim 6, further comprising the step of:

computing a secure one way hash function of carrier signal data in the digital signal, wherein the hash function is insensitive to changes introduced into the carrier signal for the purpose of carrying the transfer function-based mask set.

20. A method for protecting a digital signal, the digital signal including digital samples in a file format having an inherent granularity, comprising the steps of:

creating a predetermined key comprised of a transfer function-based mask set that can manipulate data at the inherent granularity of the file format of the underlying digitized samples;

authenticating the predetermined key containing the correct transfer function-based mask set during playback of the data; and

metering the playback of the data to monitor content.

21. The method of claim 20, wherein the predetermined key is authenticated to authenticate message information

22. A method to prepare for the scrambling of a sample stream of data, comprising the steps of:

generating a plurality of mask sets to be used for encoding, including a random primary mask, a random convolution mask and a random start of message delimiter;

obtaining a transfer function to be implemented;

generating a message bit stream to be encoded;

loading the message bit stream, a stega-cipher map truth table, the primary mask, the convolution mask and the start of message delimiter into memory;

initializing the state of a primary mask index, a convolution mask index, and a message bit index; and

setting a message size equal to the total number of bits in the message bit stream.

23. A method to prepare for the encoding of stega-cipher information into a sample stream of data, comprising the steps of:

generating a mask set to be used for encoding, the set including a random primary mask, a random convolution mask, and a random start of message delimiter;

obtaining a message to be encoded;

compressing and encrypting the message if desired;

generating a message bit stream to be encoded;

loading the message bit stream, a stega-cipher map truth table, the primary mask, the convolution mask and the start of message delimiter into memory;

initializing the state of a primary mask index, a convolution mask index, and a message bit index; and

setting the message size equal to the total number of bits in the message bit stream.

24. The method of claim 23 wherein the sample stream of data has a plurality of windows, further comprising the steps of:

calculating over which windows in the sample stream the message will be encoded;

computing a secure one way hash function of the information in the calculated windows, the hash function generating hash values insensitive to changes in the samples induced by a stega-cipher; and

encoding the computed hash values in an encoded stream of data.

25. The method of claim 13, wherein said step of selecting comprises the steps of:

collecting a series of random bits derived from keyboard latency intervals in random typing;

processing the initial series of random bits through an MD5 algorithm;

using the results of the MD5 processing to seed a triple-DES encryption loop;

cycling through the triple-DES encryption loop, extracting the least significant bit of each result after each cycle; and

concatenating the triple-DES output bits into the random series of bits.

26. A method for copy protection of digital information, the digital information including a digital sample and format information, comprising the steps of:

identifying a portion of the digital sample to be encoded;

generating an encoded digital sample from the identified portion of the digital sample; and

generating encoded digital information, including the encoded digital sample and the format information.

27. The method of claim 26, further comprising the step of requiring a predetermined key to decode the encoded digital sample.

28. The method of claim 27, wherein the digital sample and format information are configured to be used with a digital player, and wherein information output from the digital player will have a degraded quality unless the encoded digital sample is decoded with the predetermined key.

29. The method of claim 27, wherein information output will have non authentic message data unless the encode digital sample is decoded with the predetermined key.

ABSTRACT OF THE DISCLOSURE

A method for combining transfer functions with predetermined key creation. In one embodiment, digital information, including a digital sample and format information, is protected by identifying and encoding a portion of the format information. Encoded digital information, including the digital sample and the encoded format information, is generated to protect the original digital information. In another embodiment, a digital signal, including digital samples in a file format having an inherent granularity, is protected by creating a predetermined key. The predetermined key is comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

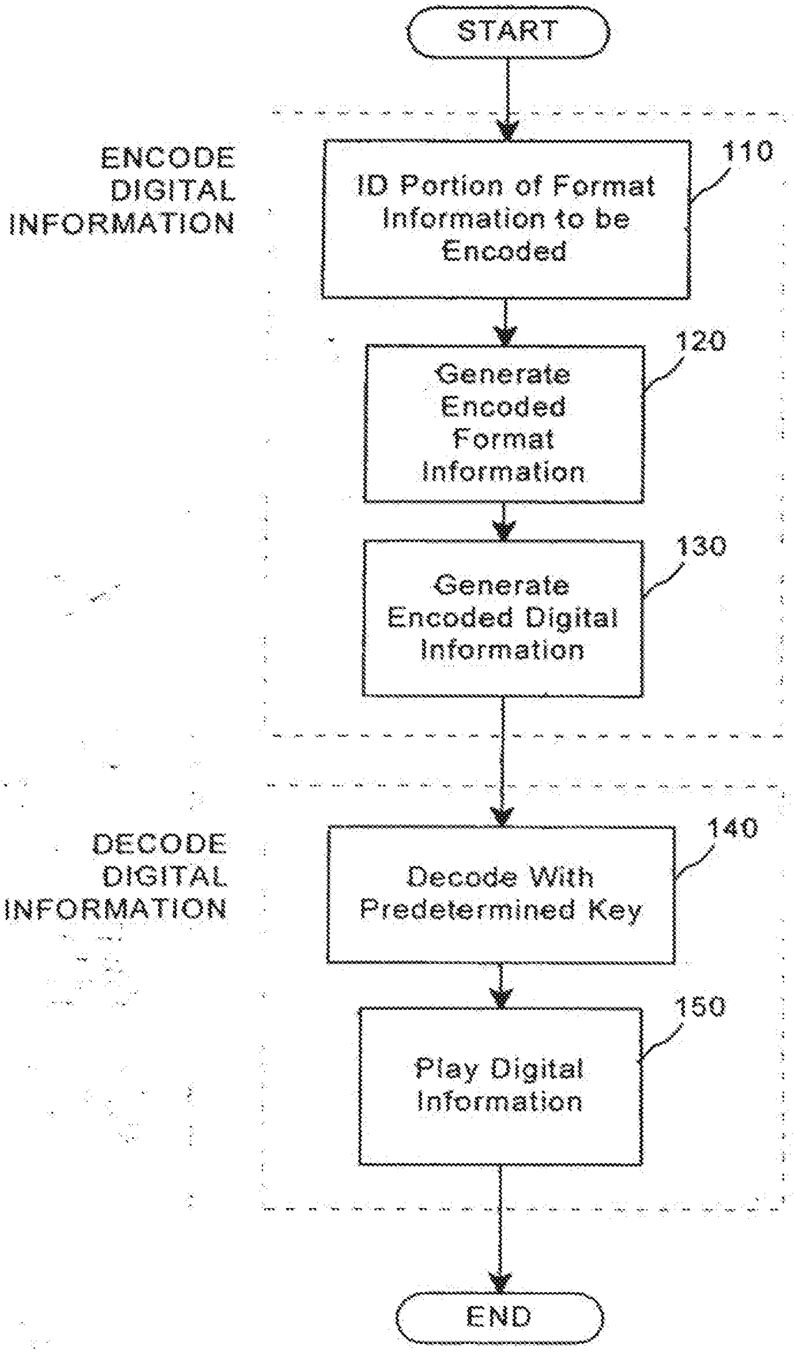

START
ENCODE
DIGITAL
INFORMATION
ID Portion of Format Information to be Encoded
110
Generate Encoded Format Information
120
Generate Encoded Digital Information
130
DECODE
DIGITAL
INFORMATION
Decode With Predetermined Key
140
Play Digital Information
150
END

FIG. 1

Attaachment 14 Page 46 of 51

Attorney Docket No: 80391.0003/Con

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As one of the below named inventors, I hereby declare that:

My residence, post office address and citizenship is as stated below next to my name;

I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter that is claimed and for which a patent is sought on the invention entitled:

METHOD FOR COMBINING TRANSFER FUNCTIONS WITH PREDETERMINED KEY CREATION

the specification of which: ☒ is attached hereto.
□ was filed on: ____________
as Application No.: ____________
and was amended on: ____________.

I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information that is material to patentability as defined in 37 C.F.R. § 1.56.

And I hereby authorize and request our agents, Wiley Rein & Fielding LLP, whose address is set forth below, to insert above, the filing date and application number of said application when known.

Prior Foreign Application(s)

I hereby claim foreign priority benefits under Title 35, United States Code, § 119(a)-(d) or § 365(b) of any foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed	
				Yes □	No □
				Yes □	No □

Prior Provisional Application(s)

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below:

Application Number	Date of Filing (day, month, year)

Prior United States Application(s)

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s), or § 365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Number	Date of Filing (day, month, year)	Status – Patented, Pending, Abandoned
09/046,627	3/24/1998	Pending

I hereby appoint, both jointly and severally, as my attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith the following attorneys, their registration numbers being listed after their names:

Floyd B. Chapman, Registration No. 40,555; David J. Kulik, Registration No. 36,576; James T. Bruce, III, Registration No. 31,491; Gregory R. Lyons, Registration No. 37,666; James H. Wallace, Jr., Registration No. 25,541; Kristin Yohannan, Registration No. 38,665; Kevin Anderson, Registration No. 43,471; Scott Bain, Registration No. 46,357; Kristin Davis, Registration No. 51,599; Christopher Hale, Registration No. 48,940; John Kuzin, Registration No. 46,848; Christopher Mills, Registration No. 46,934; Mark Pacella, Registration No. 46,974; and David Walker, Registration No. 43,976, all of

Wiley Rein & Fielding LLP, 1776 K Street, N.W., Washington, D.C., 20006, associated with **Customer Number 29693**

Attorney Docket No: 80391.0003/Con

All correspondence and telephone communications should be addressed to:

Floyd Chapman, Esq
Wiley Rein & Fielding LLP
Attn: Patent Administration
1776 K Street, N.W.
Washington, D.C. 20006

Telephone Number: 202.719.7000
Facsimile Number: 202.719.7049

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine and imprisonment, or both, under 18 U.S.C. § 1001, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature ______________________ Date 6/24/03

Full Name of
First Inventor: MOSKOWITZ (Family Name) SCOTT (First Given Name) A. (Second Given Name)

Citizenship: USA

Residence: 16711 Collins Avenue, #2505, Miami, FL 33160

Post Office
Address: Same

WRFMAIN 12090307.1

PATENT APPLICATION SERIAL NO. ____________________

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

```
06/26/2003 STEUMEL1 00000067 501129   10602777

01 FC:2001          375.00 DA
02 FC:2202            9.00 DA
03 FC:2201           84.00 DA
```

PTO-1556
(5/87)

PATENT APPLICATION FEE DETERMINATION RECORD

Effective January 1, 2003

Application or Docket Number

80391,0003

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	21	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	21 minus 20=	* 1
INDEPENDENT CLAIMS	5 minus 3 =	* 2
MULTIPLE DEPENDENT CLAIM PRESENT		□

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE □ OR OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
BASIC FEE	375.00	OR	BASIC FEE	750.00
X$ 9=	9	OR	X$18=	
X42=	84	OR	X84=	
+140=	0	OR	+280=	
TOTAL	468	OR	TOTAL	

CLAIMS AS AMENDED - PART II

SMALL ENTITY OR OTHER THAN SMALL ENTITY

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

PTO/SB/05 (07-07)
Approved for use through 06/30/2010. OMB 0651-0032
U.S. Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	80391.0003CONT2
First Inventor	Scott MOSKOWITZ
Title	Data Protection Method and Device
Express Mail Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: **Commissioner for Patents**
P.O. Box 1450
Alexandria VA 22313-1450

1. ☑ **Fee Transmittal Form** (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)
2. ☑ **Applicant claims small entity status.**
See 37 CFR 1.27.
3. ☑ **Specification** *[Total Pages ____________]*
Both the claims and abstract must start on a new page
(For information on the preferred arrangement, see MPEP 608.01(a))
4. ☑ **Drawing(s)** *(35 U.S.C. 113)* *[Total Sheets 1]*
5. **Oath or Declaration** *[Total Sheets ____________]*
a. ☑ Newly executed (original or copy)
b. □ A copy from a prior application (37 CFR 1.63(d))
(for continuation/divisional with Box 18 completed)
i. □ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) name in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. ☑ **Application Data Sheet.** See 37 CFR 1.76
7. □ **CD-ROM or CD-R** in duplicate, large table or Computer Program *(Appendix)*
□ Landscape Table on CD
8. **Nucleotide and/or Amino Acid Sequence Submission**
(if applicable, items a. – c. are required)
a. □ Computer Readable Form (CRF)
b. Specification Sequence Listing on:
i. □ CD-ROM or CD-R (2 copies); or
ii. □ Paper
c. □ Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

9. □ **Assignment Papers** (cover sheet & document(s))
Name of Assignee ______________________________
____
10. □ **37 CFR 3.73(b) Statement** *(when there is an assignee)* □ **Power of Attorney**
11. □ **English Translation Document** *(if applicable)*
12. □ **Information Disclosure Statement** (PTO/SB/08 or PTO-1449)
□ Copies of citations attached
13. ☑ **Preliminary Amendment**
14. ☑ **Return Receipt Postcard** (MPEP 503)
(Should be specifically itemized)
15. □ **Certified Copy of Priority Document(s)**
(if foreign priority is claimed)
16. □ **Nonpublication Request** under 35 U.S.C. 122(b)(2)(B)(i).
Applicant must attach form PTO/SB/35 or equivalent.
17. □ Other: ______________________________
____

18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in the first sentence of the specification following the title, or in an Application Data Sheet under 37 CFR 1.76:

□ Continuation ☑ Divisional □ Continuation-in-part (CIP) of prior application No.: ....

Prior application information: Examiner Laurel LASHLEY Art Unit: 2132

19. CORRESPONDENCE ADDRESS

□ The address associated with Customer Number: [ ] **OR** ☑ Correspondence address below

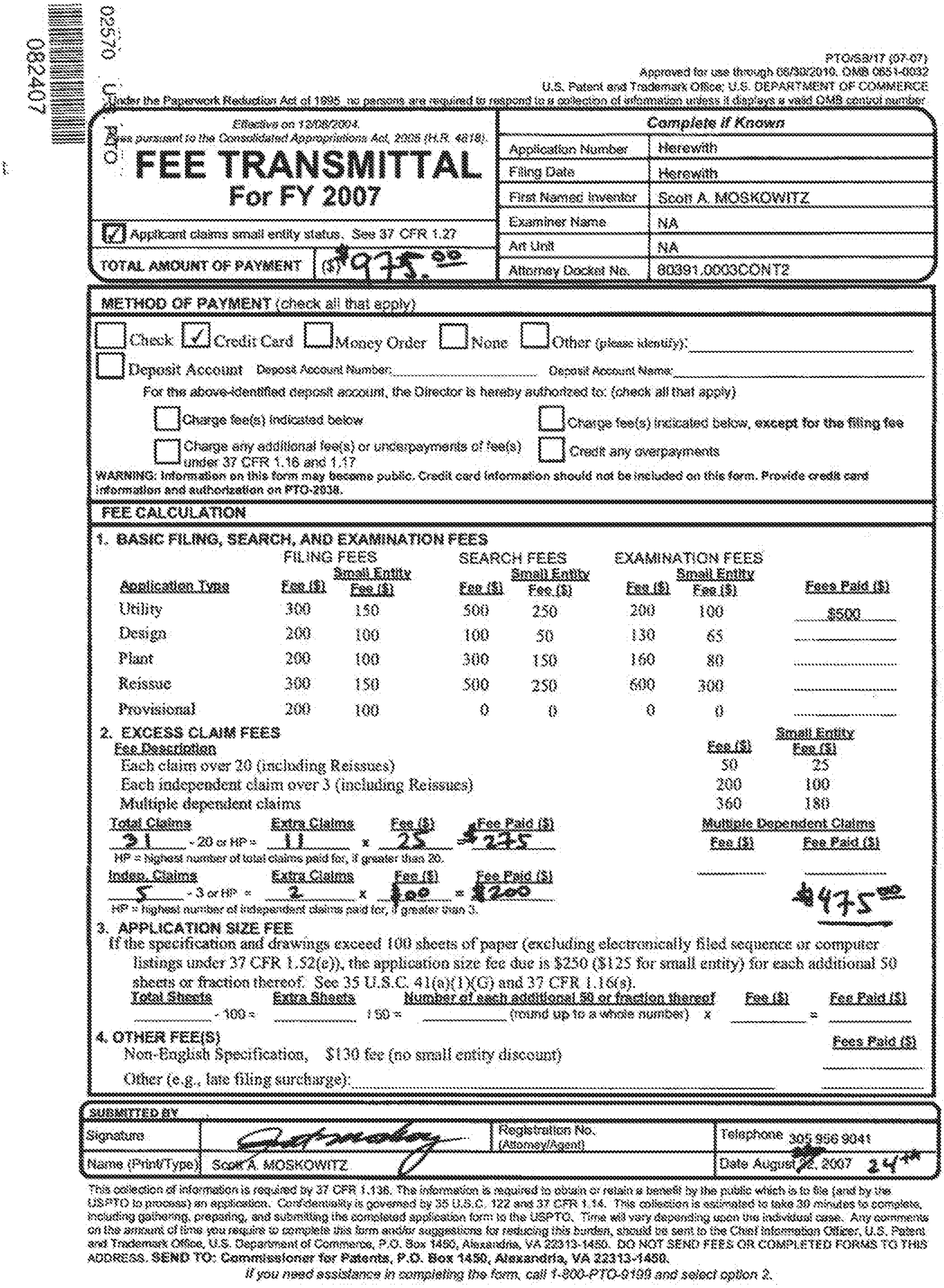
Name	Scott MOSKOWITZ				
Address	16711 Collins Avenue, #2505				
City	Sunny Isles Beach	State	FL	Zip Code	33160
Country	USA	Telephone	305 956 9041	Email	

Signature	Scott Moskowitz	Date	August , 2007 AUGUST 24, 2007
Name (Print/Type)	Scott MOSKOWITZ	Registration No. (Attorney/Agent)	

This collection of information is required by 37 CFR 1.53(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Attachment 15 Page 1 of 104



PTO/SB/17 (07-07)
Approved for use through 08/30/2010. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2007

☑ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) $975.00

Complete if Known	
Application Number	Herewith
Filing Date	Herewith
First Named Inventor	Scott A. MOSKOWITZ
Examiner Name	NA
Art Unit	NA
Attorney Docket No.	80391.0003CONT2

METHOD OF PAYMENT (check all that apply)

□ Check ☑ Credit Card □ Money Order □ None □ Other (please identify): ____________

□ Deposit Account Deposit Account Number: ____________ Deposit Account Name: ____________

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

□ Charge fee(s) indicated below □ Charge fee(s) indicated below, **except for the filing fee**

□ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 □ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
Application Type	**Fee ($)**	**Small Entity Fee ($)**	**Fee ($)**	**Small Entity Fee ($)**	**Fee ($)**	**Small Entity Fee ($)**	**Fees Paid ($)**
Utility	300	150	500	250	200	100	$500
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	**Fee ($)**	**Small Entity Fee ($)**
Each claim over 20 (including Reissues)	50	25
Each independent claim over 3 (including Reissues)	200	100
Multiple dependent claims	360	180

Total Claims 31 - 20 or HP = **Extra Claims** 11 x **Fee ($)** 25 = **Fee Paid ($)** $275

HP = highest number of total claims paid for, if greater than 20.

Indep. Claims 5 - 3 or HP = **Extra Claims** 2 x **Fee ($)** $100 = **Fee Paid ($)** $200

HP = highest number of independent claims paid for, if greater than 3.

Multiple Dependent Claims
Fee ($) ________ **Fee Paid ($)** ________

$475.00

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets ________ - 100 = **Extra Sheets** ________ / 50 = **Number of each additional 50 or fraction thereof** ________ (round up to a whole number) x **Fee ($)** ________ = **Fee Paid ($)** ________

4. OTHER FEE(S) **Fees Paid ($)**

Non-English Specification, $130 fee (no small entity discount) ________

Other (e.g., late filing surcharge): ____________________ ________

SUBMITTED BY

Signature		Registration No. (Attorney/Agent)	Telephone 305 956 9041
Name (Print/Type)	Scott A. MOSKOWITZ		Date August ~~22~~, 2007 24th

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

DATA PROTECTION METHOD AND DEVICE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a divisional of U.S. Patent Application Serial No. 10/602,777, which is a continuation application of U.S. Patent Application Serial No. 09/046,627 (which issued July 22, 2003, as U.S. Patent No. 6,598,162), which is a continuation-in-part of U.S. Patent Application Serial No. 08/587,943, filed Jan. 17, 1996, (which issued April 28, 1998, as U.S. Patent No. 5,745,943). The entire disclosure of U.S. Patent Application No. 09/046,627 (which issued July 22, 2003, as U.S. Patent No. 6,598,162) and U.S. Patent Application Serial No. 08/587,943, filed Jan. 17, 1996, (which issued April 28, 1998, as U.S. Patent No. 5,745,943) are hereby incorporated by reference in their entireties.

FIELD OF THE INVENTION

[0002] The invention relates to the protection of digital information. More particularly, the invention relates to a method and device for data protection.

[0003] With the advent of computer networks and digital multimedia, protection of intellectual property has become a prime concern for creators and publishers of digitized copies of copyrightable works, such as musical recordings, movies, video games, and computer software. One method of protecting copyrights in the digital domain is to use "digital watermarks. "

[0004] The prior art includes copy protection systems attempted at many stages in the development of the software industry. These may be various methods by which a software engineer can write the software in a clever manner to determine if it has been copied, and if so to deactivate itself. Also included are undocumented changes to the storage format of the content. Copy protection was generally abandoned by the software industry, since pirates were generally just as clever as the software engineers and figured out ways to modify the software and deactivate the protection. The cost of developing such protection was not justified considering the level of piracy which occurred despite the copy protection.

[0005] Other methods for protection of computer software include the requirement of entering certain numbers or facts that may be included in a packaged software's manual, when prompted at start-up. These may be overcome if copies of the manual are distributed to unintended users, or by patching the code to bypass these measures. Other methods include requiring a user to contact the software vendor and to receive "keys" for unlocking software after registration attached to some payment scheme, such as credit card authorization. Further methods include network-based searches of a user's hard drive and comparisons between what is registered to that user and what is actually installed on the user's general computing device. Other proposals, by such parties as AT&T's Bell Laboratories, use "kerning" or actual distance in pixels, in the rendering of text documents, rather than a varied number of ASCII characters. However, this approach can often be defeated by graphics processing analogous to sound processing, which randomizes that information. All of these methods require outside determination and verification of the validity of the software license.

[0006] Digital watermarks can be used to mark each individual copy of a digitized work with information identifying the title, copyright holder, and even the licensed owner of a particular copy. When marked with licensing and ownership information, responsibility is created for individual copies where before there was none. Computer application programs can be watermarked by watermarking digital content resources used in conjunction with images or audio data. Digital watermarks can be encoded with random or pseudo random keys, which act as secret maps for locating the watermarks. These keys make it impossible for a party to find the watermark without having the key. In addition, the encoding method can be enhanced to force a party to cause damage to a watermarked data stream when trying to erase a random-key watermark. Other information is disclosed in "Technology: Digital Commerce", Denise Caruso, New York Times, Aug. 7, 1995; and "Copyrighting in the Information Age", Harley Ungar, ONLINE MARKETPLACE, September 1995, Jupiter Communications.

[0007] Additionally, other methods for hiding information signals in content signals, are disclosed in U.S. Pat. No. 5,319,735--Preuss et al. and U.S. Pat. No. 5,379,345--Greenberg.

[0008] It is desirable to use a "stega-cipher" or watermarking process to hide the necessary parts or resources of the executable object code in the digitized sample resources. It is also desirable to further modify the underlying structure of an executable computer application such that it is more resistant to attempts at patching and analysis by memory capture. A computer application seeks to provide a user with certain utilities or tools, that is, users interact with a computer or similar device to accomplish various tasks and applications provide the relevant interface. Thus, a level of authentication can also be introduced into software, or "digital products," that include digital content, such as audio, video, pictures or multimedia, with digital watermarks. Security is maximized because erasing this code watermark without a key results in the destruction of one or more essential parts of the underlying application, rendering the "program" useless to the unintended user who lacks the appropriate key. Further, if the key is linked to a license code by means of a mathematical function, a mechanism for identifying the licensed owner of an application is created.

[0009] It is also desirable to randomly reorganize program memory structure intermittently during program run time, to prevent attempts at memory capture or object code analysis aimed at eliminating licensing or ownership information, or otherwise modifying, in an unintended manner, the functioning of the application.

[0010] In this way, attempts to capture memory to determine underlying functionality or provide a "patch" to facilitate unauthorized use of the "application," or computer program, without destroying the functionality and thus usefulness of a copyrightable computer program can be made difficult or impossible.

[0011] It is thus the goal of the present invention to provide a higher level of copyright security to object code on par with methods described in digital watermarking systems for digitized media content such as pictures, audio, video and multimedia content in its multifarious forms, as described in previous disclosures, "Steganographic Method and Device" Ser. No. 08/489,172, filed Jun. 7, 1995, now U.S. Pat. No. 5,613,004, and "Human Assisted Random Key Generation and Application for Digital Watermark System", Ser. No. 08/587,944, filed on Jan. 17, 1996, the disclosure of which is hereby incorporated by reference.

[0012] It is a further goal of the present invention to establish methods of copyright protection that can be combined with such schemes as software metering, network distribution of code and specialized protection of software that is designed to work over a network, such as that proposed by Sun Microsystems in their HotJava browser and Java programming language, and manipulation of application code in proposed distribution of documents that can be exchanged with resources or the look and feel of the document being preserved over a network. Such systems are currently being offered by companies including Adobe, with their Acrobat software. This latter goal is accomplished primarily by means of the watermarking of font, or typeface, resources included in applications or documents, which determine how a bitmap representation of the document is ultimately drawn on a presentation device.

[0013] The present invention includes an application of the technology of "digital watermarks." As described in previous disclosures, "Steganographic Method and Device" and "Human Assisted Random Key Generation and Application for Digital Watermark System," watermarks are particularly suitable to the identification, metering, distributing and authenticating digitized content such as pictures, audio, video and derivatives thereof under the description of "multimedia content." Methods have been described for combining both cryptographic methods, and steganography, or hiding something in plain view. Discussions of these technologies can be found in Applied Cryptography by Bruce Schneier and The Code Breakers by David Kahn. For more information on prior art public-key cryptosystems see U.S. Pat. No. 4,200,770 Diffie-Hellman, U.S. Pat. No. 4,218,582 Hellman, U.S. Pat. No. 4,405,829 RSA, U.S. Pat. No. 4,424,414 Hellman Pohlig. Computer code, or machine language instructions, which are not digitized and have zero tolerance for error, must be protected by derivative or alternative methods, such as those disclosed in this invention, which focuses on watermarking with "keys" derived from license codes or other ownership identification information, and using the watermarks encoded with such keys to hide an essential subset of the application code resources.

BACKGROUND OF THE INVENTION

[0014] Increasingly, commercially valuable information is being created and stored in "digital" form. For example, music, photographs and video can all be stored and transmitted as a series of numbers, such as 1's and 0's. Digital techniques let the original information be recreated in a very accurate manner. Unfortunately, digital techniques also let the information be easily copied without the information owner's permission.

[0015] Because unauthorized copying is clearly a disincentive to the digital distribution of valuable information, it is important to establish responsibility for copies and derivative copies of such works. For example, if each authorized digital copy of a popular song is identified with a unique number, any unauthorized copy of the song would also contain the number. This would allow the owner of the information, such as a song publisher, to investigate who made the unauthorized copy. Unfortunately, it is possible that the unique number could be erased or altered if it is simply tacked on at the beginning or end of the digital information.

[0016] As will be described, known digital "watermark" techniques give creators and publishers of digitized multimedia content localized, secured identification and authentication of that content. In considering the various forms of multimedia content, such as "master," stereo, National Television Standards Committee (NTSC) video, audio tape or compact disc, tolerance of quality will vary with individuals and affect the underlying commercial and aesthetic value of the content. For example, if a digital version of a popular song sounds distorted, it will be less valuable to users. It is therefore desirable to embed copyright, ownership or purchaser information, or some combination of these and related data, into the content in a way that will damage the content if the watermark is removed without authorization.

[0017] To achieve these goals, digital watermark systems insert ownership information in a way that causes little or no noticeable effects, or "artifacts," in the underlying content signal. For example, if a digital watermark is inserted into a digital version of a song, it is important that a listener not be bothered by the slight changes introduced by the watermark. It is also important for the watermark technique to maximize the encoding level and "location sensitivity" in the signal to force damage to the content signal when removal is attempted. Digital watermarks address many of these concerns, and research in the field has provided extremely robust and secure implementations.

[0018] What has been overlooked in many applications described in the art, however, are systems which closely mimic distribution of content as it occurs in the real world. For instance, many watermarking systems require the original un-watermarked content signal to enable detection or decode operations. These include highly publicized efforts by NEC, Digimarc and others. Such techniques are problematic because, in the real world, original master copies reside in a rights holders vaults and are not readily available to the public.

[0019] With much activity overly focused on watermark survivability, the security of a digital watermark is suspect. Any simple linear operation for encoding information into a signal may be used to erase the embedded signal by inverting the process. This is not a difficult task, especially when detection software is a plug-in freely available to the public, such as with Digimarc. In general, these systems seek to embed cryptographic information, not cryptographically embed information into target media content.

[0020] Other methods embed ownership information that is plainly visible in the media signal, such as the method described in U.S. Pat. No. 5,530,739 to Braudaway et al. The system described in Braudaway protects a digitized image by encoding a visible watermark to deter piracy. Such an implementation creates an immediate weakness in securing the embedded information because the watermark is plainly visible. Thus, no search for the embedded signal is necessary and the watermark can be more easily removed or altered. For example, while certainly useful to some rights owners, simply placing the symbol "©" in the digital information would only provide limited protection. Removal by adjusting the brightness of the pixels forming the "©" would not be difficult with respect to the computational resources required.

[0021] Other relevant prior art includes U.S. Pat. No. 4,979,210 and 5,073,925 to Nagata et al., which encodes information by modulating an audio signal in the amplitude/time domain. The modulations introduced in the Nagata process carry a "copy/don't copy" message, which is easily found and circumvented by one skilled in the art. The granularity of encoding is fixed by the amplitude and frequency modulation limits required to maintain inaudibility. These limits are relatively low, making it impractical to encode more information using the Nagata process.

[0022] Although U.S. Pat. No. 5,661,018 to Leighton describes a means to prevent collusion attacks in digital watermarks, the disclosed method may not actually provide the security described. For-example, in cases where the watermarking technique is linear, the "insertion envelope" or "watermarking space" is well-defined and thus susceptible to attacks less sophisticated than collusion by unauthorized parties. Over-encoding at the watermarking encoding level is but one simple attack in such linear implementations. Another consideration not made by Leighton is that commercially-valuable content may already exist in a un-watermarked form somewhere, easily accessible to potential pirates, gutting the need for any type of collusive activity. Digitally signing the embedded signal with preprocessing of watermark data is more likely to prevent successful collusion. Furthermore, a "baseline" watermark as disclosed is quite subjective. It is simply described elsewhere in the art as the "perceptually significant" regions of a signal. Making a watermarking function less linear or inverting the insertion of watermarks would seem to provide the same benefit without the additional work required to create a "baseline" watermark. Indeed, watermarking algorithms should already be capable of defining a target insertion envelope or region without additional steps. What is evident is the Leighton patent does not allow for initial prevention of attacks on an embedded watermark as the content is visibly or audibly unchanged.

[0023] It is also important that any method for providing security also function with broadcasting media over networks such as the Internet, which is also referred to as "streaming." Commercial "plug-in" products such as RealAudio and RealVideo, as well as applications by vendors VDONet and Xtreme, are common in such network environments. Most digital watermark implementations focus on common file base signals and fail to anticipate the security of streamed signals. It is desirable that any protection scheme be able to function with a plug-in player without advanced knowledge of the encoded media stream.

[0024] Other technologies focus solely on file-based security. These technologies illustrate the varying applications for security that must be evaluated for different media and distribution environments. Use of cryptolopes or cryptographic containers, as proposed by IBM in its Cryptolope product, and InterTrust, as described in U.S. Pat. Nos. 4,827,508, 4,977,594, 5,050,213 and 5,410,598, may discourage certain forms of piracy. Cryptographic containers, however, require a user to subscribe to particular decryption software to decrypt data. IBM's InfoMarket and InterTrust's DigiBox, among other implementations, provide a generalized model and need proprietary architecture to function. Every user must have a subscription or registration with the party which encrypts the data. Again, as a form of general encryption, the data is scrambled or encrypted without regard to the media and its formatting. Finally, control over copyrights or other neighboring rights is left with the implementing party, in this case, IBM, InterTrust or a similar provider.

[0025] Methods similar to these "trusted systems" exist, and Cerberus Central Limited and Liquid Audio, among a number of companies, offer systems which may functionally be thought of as subsets of IBM and InterTrust's more generalized security offerings. Both Cerberus and Liquid Audio propose proprietary player software which is registered to the user and "locked" in a manner parallel to the locking of content that is distributed via a cryptographic container. The economic trade-off in this model is that users are required to use each respective companies' proprietary player to play or otherwise manipulate content that is downloaded. If, as is the case presently, most music or other media is not available via these proprietary players and more companies propose non-compatible player formats, the proliferation of players will continue. Cerberus and Liquid Audio also by way of extension of their architectures provide for "near-CD quality" but proprietary compression. This requirement stems from the necessity not to allow content that has near-identical data make-up to an existing consumer electronic standard, in Cerberus and Liquid Audio's case the so-called Red Book audio CD standard of 16 bit 44.1 kHz, so that comparisons with the proprietary file may not yield how the player is secured. Knowledge of the player's file format renders its security ineffective as a file may be replicated and played on any common player, not the intended proprietary player of the provider of previously secured and uniquely formatted content. This is the parallel weakness to public key crypto-systems which have gutted security if enough plain text and cipher text comparisons enable a pirate to determine the user's private key.

[0026] Many approaches to digital watermarking leave detection and decoding control with the implementing party of the digital watermark, not the creator of the work to be protected. A set of secure digital watermark implementations address this fundamental control issue forming the basis of key-based approaches. These are covered by the following patents and pending applications, the entire disclosures of which are hereby incorporated by reference: U.S. Pat. No. 5,613, 004 entitled "Steganographic Method and Device" and its derivative U.S. patent application Ser. No. 08/775,216 (which issued November 11, 1997, as U.S. Patent No. 5,687,236), U.S. patent application Ser. No. 08/587,944 entitled "Human Assisted Random Key Generation and Application for Digital Watermark System"(which issued October 13, 1998, as U.S. Patent No. 5,822,432), U.S. patent application Ser. No. 08/587,943 entitled "Method for Stega-Cipher Protection of Computer Code"(which issued April 28, 1998, as U.S. Patent No. 5,748,569), U.S. patent application Ser. No. 08/677,435 entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data"(which issued March 30, 1999, as U.S. Patent No. 5,889,868) and U.S. patent application Ser. No. 08/772,222 entitled "Z-Transform Implementation of Digital Watermarks"(which issued June 20, 2000, as U.S. Patent No. 6,078,664). Public key crypto-systems are described in U.S. Pat. No. 4,200,770, 4,218,582, 4,405,829 and 4,424,414, the entire disclosures of which are also hereby incorporated by reference.

[0027] In particular, an improved protection scheme is described in "Method for Stega-Cipher Protection of Computer Code," U.S. patent application Ser. No. 08/587,943 (which issued April 28, 1998, as U.S. Patent No. 5,748,569). This technique uses the key-based insertion of binary executable computer code within a content signal that is subsequently, and necessarily, used to play or otherwise manipulate the signal in which it is encoded. With this system, however, certain computational requirements, such as one digital player per digital copy of content, may be necessitated. For instance, a consumer may download many copies of watermarked content. With this technique, the user would also be downloading as many copies of the digital player program. While this form of security may be desirable for some applications, it is not appropriate in many circumstances.

[0028] Finally, even when digital information is distributed in encoded form, it may be desirable to allow unauthorized users to play the information with a digital player, perhaps with a reduced level of quality. For example, a popular song may be encoded and freely distributed in encoded form to the public. The public, perhaps using commonly available plug-in digital players, could play the encoded content and hear the music in some degraded form. The music may-sound choppy, or fuzzy or be degraded in some other way. This lets the public decide, based on the available lower quality version of the song, if they want to purchase a key from the publisher to decode, or "clean-up," the content. Similar approaches could be used to distribute blurry pictures or low quality video. Or even "degraded" text, in the sense that only authenticated portions of the text can be determined with the predetermined key or a validated digital signature for the intended message.

[0029] In view of the foregoing, it can be appreciated that a substantial need exists for a method allowing encoded content to be played, with degraded quality, by a plug-in digital player, and solving the other problems discussed above.

SUMMARY OF THE INVENTION

[0030] The disadvantages of the art are alleviated to a great extent by a method for combining transfer functions with predetermined key creation. In one embodiment, digital information, including a digital sample and format information, is protected by identifying and encoding a portion of the format information. Encoded digital information, including the digital sample and the encoded format information, is generated to protect the original digital information.

[0031] In another embodiment, a digital signal, including digital samples in a file format having an inherent granularity, is protected by creating a predetermined key. The predetermined key is comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

[0032] It is thus a goal of the present invention, to provide a level of security for executable code on similar grounds as that which can be provided for digitized samples. Furthermore, the present invention differs from the prior art in that it does not attempt to stop copying, but rather, determines responsibility for a copy by ensuring that licensing information must be preserved in descendant copies from an original. Without the correct license information, the copy cannot function.

[0033] An improvement over the art is disclosed in the present invention, in that the software itself is a set of commands, compiled by software engineer, which can be configured in such a manner as to tie underlying functionality to the license or authorization of the copy in possession by the user. Without such verification, the functions sought out by the user in the form of software cease to properly work. Attempts to tamper or "patch" substitute code resources can be made highly difficult by randomizing the location of said resources in memory on an intermittent basis to resist most attacks at disabling the system.

[0034] With these and other advantages and features of the invention that will become hereinafter apparent, the nature of the invention may be more clearly understood by reference to the following detailed description of the invention, the appended claims and to the several drawings attached herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0035] FIG. 1 is a block flow diagram of a method for copy protection or authentication of digital information according to an embodiment of the present invention.

DETAILED DESCRIPTION

[0036] In accordance with an embodiment of the present invention, a method combines transfer functions with predetermined key creation. Increased security is achieved in the method by combining elements of "public-key steganography" with cryptographic protocols, which keep in-transit data secure by scrambling the data with "keys" in a manner that is not apparent to those with access to the content to be distributed. Because different forms of randomness are combined to offer robust, distributed security, the present invention addresses an architectural "gray space" between two important areas of security: digital watermarks, a subset of the more general art of steganography, and cryptography. One form of randomness exists in the mask sets that are randomly created to map watermark data into an otherwise unrelated digital signal. The second form of randomness is the random permutations of data formats used with digital players to manipulate the content with the predetermined keys. These forms can be thought of as the transfer function versus the mapping function inherent to digital watermarking processes.

[0037] According to an embodiment of the present invention, a predetermined, or randomly generated, key is used to scramble digital information in a way that is unlike known "digital watermark" techniques and public key crypto-systems. As used herein, a key is also referred to as a "mask set" which includes one or more random or pseudo-random series of bits. Prior to encoding, a mask can be generated by any cryptographically secure random generation process. A block cipher, such as a Data Encryption Standard (DES) algorithm, in combination with a sufficiently random seed value, such as one created using a Message Digest 5 (MD5) algorithm, emulates a cryptographically secure random bit generator. The keys are saved in a database, along with information matching them to the digital signal, for use in descrambling and subsequent viewing or playback. Additional file format or transfer property information is prepared and made available to the encoder, in a bit addressable manner. As well, any authenticating function can be combined, such as Digital Signature Standard (DSS) or Secure Hash Algorithm (SHA).

[0038] Using the predetermined key comprised of a transfer function-based mask set, the data representing the original content is manipulated at the inherent granularity of the file format of the underlying digitized samples. Instead of providing, or otherwise distributing, watermarked content that is not noticeably altered, a partially "scrambled" copy of the content is distributed. The key is necessary both to register the sought-after content and to descramble the content into its original form.

[0039] The present invention uses methods disclosed in "Method for Stega-Cipher Protection of Computer Code," U.S. patent application Ser. No. 08/587,943 (which issued April 28, 1998, as U.S. Patent No. 5,748,569), with respect to transfer functions related to the common file formats, such as PICT, TIFF, AIFF, WAV, etc. Additionally, in cases where the content has not been altered beyond being encoded with such functional data, it is possible for a digital player to still play the content because the file format has not been altered. Thus, the encoded content could still be played by a plug-in digital player as discrete, digitally sampled signals, watermarked or not. That is, the structure of the file can remain basically unchanged by the watermarking process, letting common file format based players work with the "scrambled" content.

[0040] For example, the Compact Disc-Digital Audio (CD-DA) format stores audio information as a series of frames. Each frame contains a number of digital samples representing, for example, music, and a header that contains file format information. As shown in FIG. 1, according to an embodiment of the present invention some of the header information can be identified and "scrambled" using the predetermined key at steps 110 to 130. The music samples can remain unchanged. Using this technique, a traditional CD-DA player will be able to play a distorted version of the music in the sample. The amount of distortion will depend on the way, and extent, that the header, or file format, information has been scrambled. It would also be possible to instead scramble some of the digital samples while leaving the header information alone. In general, the digital signal would be protected by manipulating data at the inherent granularity, or "frames," of the CD-DA file format. To decode the information, a predetermined key is used before playing the digital information at steps 140 and 150.

[0041] A key-based decoder can act as a "plug-in" digital player of broadcast signal streams without foreknowledge of the encoded media stream. Moreover, the data format orientation is used to partially scramble data in transit to prevent unauthorized descrambled access by decoders that lack authorized keys. A distributed key can be used to unscramble the scrambled content because a decoder would understand how to process the key. Similar to on-the-fly decryption operations, the benefits inherent in this embodiment include the fact that the combination of watermarked content security, which is key-based, and the descrambling of the data, can be performed by the same key which can be a plurality of mask sets. The mask sets may include primary, convolution and message delimiter masks with file format data included.

[0042] The creation of an optimized "envelope" for insertion of watermarks provides the basis of much watermark security, but is also a complementary goal of the present invention. The predetermined or random key that is generated is not only an essential map to access the hidden information signal, but is also the descrambler of the previously scrambled signal's format for playback or viewing.

[0043] In a system requiring keys for watermarking content and validating the distribution of the content, different keys may be used to encode different information while secure one way hash functions or one-time pads may be incorporated to secure the embedded signal. The same keys can be used to later validate the embedded digital signature, or even fully decode the digital watermark if desired. Publishers can easily stipulate that content not only be digitally watermarked but that distributors must check the validity of the watermarks by performing digital signature-checks with keys that lack any other functionality. The system can extend to simple authentication of text in other embodiments.

[0044] Before such a market is economically feasible, there are other methods for deploying key-based watermarking coupled with transfer functions to partially scramble the content to be distributed without performing full public key encryption, i.e., a key pair is not necessarily generated, simply, a predetermined key's function is created to re-map the data of the content file in a lossless process. Moreover, the scrambling performed by the present invention may be more dependent on the file in question. Dissimilarly, encryption is not specific to any particular media but is performed on data. The file format remains unchanged, rendering the file useable by any conventional viewer/player, but the signal quality can be intentionally degraded in the absence of the proper player and key. Public-key encryption seeks to completely obscure the sensitive "plaintext" to prevent comparisons with the "ciphertext" to determine a user's private keys. Centralized encryption only differs in the utilization of a single key for both encryption and decryption making the key even more highly vulnerable to attacks to defeat the encryption process. With the present invention, a highly sought after photograph may be hazy to the viewer using any number of commonly available, nonproprietary software or hardware, without the authorized key. Similarly, a commercially valuable song may sound poor.

[0045] The benefit of some form of cryptography is not lost in the present invention. In fact, some piracy can be deterred when the target signal may be known but is clearly being protected through scrambling. What is not anticipated by known techniques, is an ala carte method to change various aspects of file formatting to enable various "scrambled states" for content to be subsequently distributed. An image may lack all red pixels or may not have any of the most significant bits activated. An audio sample can similarly be scrambled to render it less-than-commercially viable.

[0046] The present invention also provides improvements over known network-based methods, such as those used for the streaming of media data over the Internet. By manipulating file formats, the broadcast media, which has been altered to "fit" within electronic distribution parameters, such as bandwidth availability and error correction considerations, can be more effectively utilized to restrict the subsequent use of the content while in transit as well as real-time viewing or playing.

[0047] The mask set providing the transfer function can be read on a per-use basis by issuing an authorized or authenticating "key" for descrambling the signal that is apparent to a viewer or a player or possessor of the authenticating key. The mask set can be read on a per-computer basis by issuing the authorized key that is more generalized for the computer that receives the broadcast signals. Metering and subscription models become viable advantages over known digital watermark systems which assist in designating the ownership of a copy of digitized media content, but do not prevent or restrict the copying or manipulation of the sampled signal in question. For broadcast or streamed media, this is especially the case. Message authentication is also possible, though not guaranteeing the same security as an encrypted file as with general crypto systems.

[0048] The present invention thus benefits from the proprietary player model without relying on proprietary players. No new players will be necessary and existing multimedia file formats can be altered to exact a measure of security which is further increased when coupled with digital watermarks. As with most consumer markets for media content, predominant file formats exist, de facto, and corresponding formats for computers likewise exist. For a commercial compact disc quality audio recording, or 16 bit 44.1 kHz, corresponding file formats include: Audio Interchange File Format (AIFF), Microsoft WAV, Sound Designer II, Sun's .au, Apple's Quicktime, etc. For still image media, formats are similarly abundant: TIFF, PICT, JPEG, GIF, etc. Requiring the use of additional proprietary players, and their complementary file formats, for limited benefits in security is wasteful. Moreover, almost all computers today are multimedia-capable, and this is increasingly so with the popularity of Intel's MMX chip architecture and the PowerPC line of microchips. Because file formatting is fundamental in the playback of the underlying data, the predetermined key can act both as a map, for information to be encoded as watermark data regarding ownership, and a descrambler of the file that has been distributed. Limitations will only exist in how large the key must be retrofitted for a given application, but any manipulation of file format information is not likely to exceed the size of data required versus that for an entire proprietary player.

[0049] As with previous disclosures by the inventor on digital watermarking techniques, the present invention may be implemented with a variety of cryptographic protocols to increase both confidence and security in the underlying system. A predetermined key is described as a set of masks. These masks may include primary, convolution and message delimiter mask. In previous disclosures, the functionality of these masks is defined solely for mapping. The present invention includes a mask set which is also controlled by the distributing party of a copy of a given media signal. This mask set is a transfer function which is limited only by the parameters of the file format in question. To increase the uniqueness or security of each key used to scramble a given media file copy, a secure one way hash function can be used subsequent to transfer properties that are initiated to prevent the forging of a particular key. Public and private keys may be used as key pairs to further increase the unlikeliness that a key may be compromised.

[0050] These same cryptographic protocols can be combined with the embodiments of the present invention in administering streamed content that requires authorized keys to correctly display or play the streamed content in an unscrambled manner. As with digital watermarking, symmetric or asymmetric public key pairs may be used in a variety of implementations. Additionally, the need for certification authorities to maintain authentic key-pairs becomes a consideration for greater security beyond symmetric key implementations. The cryptographic protocols makes possible, as well, a message of text to be authenticated by a message authenticating function in a general computing device that is able to ensure secure message exchanges between authorizing parties.

[0051] An executable computer program is variously referred to as an application, from the point of view of a user, or executable object code from the point of view of the engineer. A collection of smaller, atomic (or indivisible) chunks of object code typically comprise the complete executable object code or application which may also require the presence of certain data resources. These indivisible portions of object code correspond with the programmers' function or procedure implementations in higher level languages, such as C or Pascal. In creating an application, a programmer writes "code" in a higher level language, which is then compiled down into "machine language," or, the executable object code, which can actually be run by a computer, general purpose or otherwise. Each function, or procedure, written in the programming language, represents a self-contained portion of the larger program, and implements, typically, a very small piece of its functionality. The order in which the programmer types the code for the various functions or procedures, and the distribution of and arrangement of these implementations in various files which hold them is unimportant. Within a function or procedure, however, the order of individual language constructs, which correspond to particular machine instructions is important, and so functions or procedures are considered indivisible for purposes of this discussion. That is, once a function or procedure is compiled, the order of the machine instructions which comprise the executable object code of the function is important and their order in the computer memory is of vital importance. Note that many "compilers" perform "optimizations" within functions or procedures, which determine, on a limited scale, if there is a better arrangement for executable instructions which is more efficient than that constructed by the programmer, but does not change the result of the function or procedure. Once these optimizations are performed, however, making random changes to the order of instructions is very likely to "break" the function. When a program is compiled, then, it consists of a collection of these sub-objects, whose exact order or arrangement in memory is not important, so long as any sub-object which uses another sub-object knows where in memory it can be found.

[0052] The memory address of the first instruction in one of these sub-objects is called the "entry point" of the function or procedure. The rest of the instructions comprising that sub-object immediately follow from the entry point. Some systems may prefix information to the entry point which describes calling and return conventions for the code which follows, an example is the Apple Macintosh Operating System (MacOS). These sub-objects can be packaged into what are referred to in certain systems as "code resources," which may be stored separately from the application, or shared with other applications, although not necessarily. Within an application there are also data objects, which consist of some data to be operated on by the executable code. These data objects are not executable. That is, they do not consist of executable instructions. The data objects can be referred to in certain systems as "resources."

[0053] When a user purchases or acquires a computer program, she seeks a computer program that "functions" in a desired manner. Simply, computer software is overwhelmingly purchased for its underlying functionality. In contrast, persons who copy multimedia content, such as pictures, audio and video, do so for the entertainment or commercial value of the content. The difference between the two types of products is that multimedia content is not generally interactive, but is instead passive, and its commercial value relates more on passive not interactive or utility features, such as those required in packaged software, set-top boxes, cellular phones, VCRs, PDAs, and the like. Interactive digital products which include computer code may be mostly interactive but can also contain content to add to the interactive experience of the user or make the underlying utility of the software more aesthetically pleasing. It is a common concern of both of these creators, both of interactive and passive multimedia products, that "digital products" can be easily and perfectly copied and made into unpaid or unauthorized copies. This concern is especially heightened when the underlying product is copyright protected and intended for commercial use.

[0054] The first method of the present invention described involves hiding necessary "parts" or code "resources" in digitized sample resources using a "digital watermarking" process, such as that described in the "Steganographic Method and Device" patent application. The basic premise for this scheme is that there are a certain sub-set of executable code resources, that comprise an application and that are "essential" to the proper function of the application. In general, any code resource can be considered "essential" in that if the program proceeds to a point where it must "call" the code resource and the code resource is not present in memory, or cannot be loaded, then the program fails. However, the present invention uses a definition of "essential" which is more narrow. This is because, those skilled in the art or those with programming experience, may create a derivative program, not unlike the utility provided by the original program, by writing additional or substituted code to work around unavailable resources. This is particularly true with programs that incorporate an optional "plug-in architecture," where several code resources may be made optionally available at run-time. The present invention is also concerned with concentrated efforts by technically skilled people who can analyze executable object code and "patch" it to ignore or bypass certain code resources. Thus, for the present embodiment's purposes, "essential" means that the function which distinguishes this application from any other application depends upon the presence and use of the code resource in question. The best candidates for this type of code resources are NOT optional, or plug-in types, unless special care is taken to prevent work-arounds.

[0055] Given that there are one or more of these essential resources, what is needed to realize the present invention is the presence of certain data resources of a type which are amenable to the "stega-cipher" process described in the "Steganographic Method and Device" patent U.S. Pat. No. 5,613,004. Data which consists of image or audio samples is particularly useful. Because this data consists of digital samples, digital watermarks can be introduced into the samples. What is further meant is that certain applications include image and audio samples which are important to the look and feel of the program or are essential to the processing of the application's functionality when used by the user. These computer programs are familiar to users of computers but also less obvious to users of other devices that run applications that are equivalent in some measure of functionality to general purpose computers including, but not limited to, set-top boxes, cellular phones, "smart televisions," PDAs and the like. However, programs still comprise the underlying "operating systems" of these devices and are becoming more complex with increases in functionality.

[0056] One method of the present invention is now discussed. When code and data resources are compiled and assembled into a precursor of an executable program the next step is to use a utility application for final assembly of the executable application. The programmer marks several essential code resources in a list displayed by the utility. The utility will choose one or several essential code resources, and encode them into one or several data resources using the stegacipher process. The end result will be that these essential code resources are not stored in their own partition, but rather stored as encoded information in data resources. They are not accessible at run-time without the key. Basically, the essential code resources that provide functionality in the final end-product, an executable application or computer program, are no longer easily and recognizably available for manipulation by those seeking to remove the underlying copyright or license, or its equivalent information, or those with skill to substitute alternative code resources to "force" the application program to run as an unauthorized copy. For the encoding of the essential code resources, a "key" is needed. Such a key is similar to those described in U.S. Pat. No. 5,613,004, the "Steganographic Method and Device" patent. The purpose of this scheme is to make a particular licensed copy of an application distinguishable from any other. It is not necessary to distinguish every instance of an application, merely every instance of a license. A licensed user may then wish to install multiple copies of an application, legally or with authorization. This method, then, is to choose the key so that it corresponds, is equal to, or is a function of, a license code or license descriptive information, not just a text file, audio clip or identifying piece of information as desired in digital watermarking schemes extant and typically useful to stand-alone, digitally sampled content. The key is necessary to access the underlying code, i.e., what the user understands to be the application program.

[0057] The assembly utility can be supplied with a key generated from a license code generated for the license in question. Alternatively, the key, possibly random, can be stored as a data resource and encrypted with a derivative of the license code. Given the key, it encodes one or several essential resources into one or several data resources. Exactly which code resources are encoded into which data resources may be determined in a random or pseudo random manner. Note further that the application contains a code resource which performs the function of decoding an encoded code resource from a data resource. The application must also contain a data resource which specifies in which data resource a particular code resource is encoded. This data resource is created and added at assembly time by the assembly utility. The application can then operate as follows:

[0058] 1) when it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. This can include a particular computer configuration;

[0059] 2) it stores this information in a personalization data resource;

[0060] 3) Once it has the license code, it can then generate the proper decoding key to access the essential code resources.

[0061] Note that the application can be copied in an uninhibited manner, but must contain the license code issued to the licensed owner, to access its essential code resources. The goal of the invention, copyright protection of computer code and establishment of responsibility for copies, is thus accomplished.

[0062] This invention represents a significant improvement over prior art because of the inherent difference in use of purely informational watermarks versus watermarks which contain executable object code. If the executable object code in a watermark is essential to an application which accesses the data which contains the watermark, this creates an all-or-none situation. Either the user must have the extracted watermark, or the application cannot be used, and hence the user cannot gain full access to the presentation of the information in the watermark bearing data. In order to extract a digital watermark, the user must have a key. The key, in turn, is a function of the license information for the copy of the software in question. The key is fixed prior to final assembly of the application files, and so cannot be changed at the option of the user. That, in turn, means the license information in the software copy must remain fixed, so that the correct key is available to the software. The key and the license information are, in fact, interchangeable. One is merely more readable than the other. In U.S. Pat. No. 5,613,004, the "Steganographic Method and Device, patent", the possibility of randomization erasure attacks on digital watermarks was discussed. Simply, it is always possible to erase a digital watermark, depending on how much damage you are willing to do to the watermark-bearing content stream. The present invention has the significant advantage that you must have the watermark to be able to use the code it contains. If you erase the watermark you have lost a key piece of the functionality of the application, or even the means to access the data which bear the watermark.

[0063] A preferred embodiment would be implemented in an embedded system, with a minimal operating system and memory. No media playing "applets," or smaller sized applications as proposed in new operating environments envisioned by Sun Microsystems and the advent of Sun's Java operating system, would be permanently stored in the system, only the bare necessities to operate the device, download information, decode watermarks and execute the applets contained in them. When an applet is finished executing, it is erased from memory. Such a system would guarantee that content which did not contain readable watermarks could not be used. This is a powerful control mechanism for ensuring that content to be distributed through such a system contains valid watermarks. Thus, in such networks as the Internet or set-top box controlled cable systems, distribution and exchange of content would be made more secure from unauthorized copying to the benefit of copyright holders and other related parties. The system would be enabled to invalidate, by default, any content which has had its watermark(s) erased, since the watermark conveys, in addition to copyright information, the means to fully access, play, record or otherwise manipulate, the content.

[0064] A second method according to the present invention is to randomly re-organize program memory structure to prevent attempts at memory capture or object code analysis. The object of this method is to make it extremely difficult to perform memory capture-based analysis of an executable computer program. This analysis is the basis for a method of attack to defeat the system envisioned by the present invention.

[0065] Once the code resources of a program are loaded into memory, they typically remain in a fixed position, unless the computer operating system finds it necessary to rearrange certain portions of memory during "system time," when the operating system code, not application code, is running. Typically, this is done in low memory systems, to maintain optimal memory utilization. The MacOS for example, uses Handles, which are double-indirect pointers to memory locations, in order to allow the operating system to rearrange memory transparently, underneath a running program. If a computer program contains countermeasures against unlicensed copying, a skilled technician can often take a snapshot of the code in memory, analyze it, determine which instructions comprise the countermeasures, and disable them in the stored application file, by means of a "patch." Other applications for designing code that moves to prevent scanning-tunnelling microscopes, and similar high sensitive hardware for analysis of electronic structure of microchips running code, have been proposed by such parties as Wave Systems. Designs of Wave Systems' microchip are intended for preventing attempts by hackers to "photograph" or otherwise determine "burn in" to microchips for attempts at reverse engineering. The present invention seeks to prevent attempts at understanding the code and its organization for the purpose of patching it. Unlike systems such as Wave Systems', the present invention seeks to move code around in such a manner as to complicate attempts by software engineers to reengineer a means to disable the methods for creating licensed copies on any device that lacks "trusted hardware." Moreover, the present invention concerns itself with any application software that may be used in general computing devices, not chipsets that are used in addition to an underlying computer to perform encryption. Wave Systems' approach to security of software, if interpreted similarly to the present invention, would dictate separate microchip sets for each piece of application software that would be tamperproof. This is not consistent with the economics of software and its distribution.

[0066] Under the present invention, the application contains a special code resource which knows about all the other code resources in memory. During execution time, this special code resource, called a "memory scheduler," can be called periodically, or at random or pseudo random intervals, at which time it intentionally shuffles the other code resources randomly in memory, so that someone trying to analyze snapshots of memory at various intervals cannot be sure if they are looking at the same code or organization from one "break" to the next. This adds significant complexity to their job. The scheduler also randomly relocates itself when it is finished. In order to do this, the scheduler would have to first copy itself to a new location, and then specifically modify the program counter and stack frame, so that it could then jump into the new copy of the scheduler, but return to the correct calling frame. Finally, the scheduler would need to maintain a list of all memory addresses which contain the address of the scheduler, and change them to reflect its new location.

[0067] The methods described above accomplish the purposes of the invention--to make it hard to analyze captured memory containing application executable code in order to create an identifiable computer program or application that is different from other copies and is less susceptible to unauthorized use by those attempting to disable the underlying copyright protection system. Simply, each copy has particular identifying information making that copy different from all other copies.

[0068] Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.

What is claimed is:

1. (original) A method for copy protection of digital information, the digital information including a digital sample and format information, comprising the steps of:

identifying a portion of the format information to be encoded;

generating encoded format information from the identified portion of the format information; and

generating encoded digital information, including the digital sample and the encoded format information.

2. (original) The method of claim 1, further comprising the step of requiring a predetermined key to decode the encoded format information.

3. (original) The method of claim 2, wherein the digital sample and format information are configured to be used with a digital player, and wherein information output from the digital player will have a degraded quality unless the encoded format information is decoded with the predetermined key.

4. (original) The method of claim 3, wherein the information output from the digital player represents a still image, audio or video.

5. (original) The method of claim 3, wherein the information output represents text data to be authenticated.

Claims 6 – 31 (cancelled without prejudice to Applicant's right to seek allowance of said claims in a related application)

32. (new) A method for copy protection of software comprising: embedding the software with a watermark wherein the embedded software operates in a manner substantially the same as the software prior to the embedding step.

33. (new) The process of claim 32, wherein the step of embedding the software with a watermark increases the complexity of code analysis and/or tampering with the software.

34. (new) The process of claim 32, wherein the watermarked software queries a user for personalization information during installation of the software

35. (new) The process of claim 32, wherein the watermark is accessible with a key.

36. (new) The process of claim 35, wherein the key enables authorized use of the watermarked software.

37. (new) The process according to claim 35, wherein the key and license information are interchangeable.

38. (new) The process according to claim 32, wherein the step of embedding the software with a watermark is performed during execution of the software.

39. (new) The process according to claim 32, wherein the step of embedding the software with a watermark modifies the structure of the software being embedded.

40. (new) An article of manufacture comprising a machine readable medium, having thereon stored instructions adapted to be executed by a processor, which instructions when executed result in a process comprising: receiving potentially watermarked software; and identifying the software by extracting the watermark.

41. (new) The article of manufacture of claim 40, wherein the watermark is associated with information fixed prior to distribution of the watermarked software.

42. (new) The article of manufacture of claim 40, wherein the watermark affects functionality of the watermarked software.

43. (new) The article of manufacture of claim 40, wherein the extracted watermark enables generation of a key.

44. (new) The article of manufacture of claim 43, wherein the generated key and licensing information are associated.

45. (new) The article of manufacture of claim 40, further comprising limiting functionality of the software if the watermark cannot be extracted.

46. (new) A method for watermarking software comprising: determining the structure a plurality of code contained in the software; and configuring at least a portion of the plurality of code according to a watermarking process.

47. (new) The process of claim 46, wherein the watermarking process further comprises inserting information into the software after installation.

48. (new) The process of claim 46, wherein the watermarking process configures the at least a portion of the plurality of code according to a key.

49. (new) The process of claim 46, wherein the watermarking process increases the complexity of code analysis and/or tampering with the software.

50. (new) The process of claim 46, wherein the watermarking process is selected from the group comprising: data hiding, steganography or steganographic ciphering.

51. (new) The process of claim 46, wherein the watermarking process is applied during execution of the software.

52. (new) A system for copy protection of software comprising the steps of: associating license information with a copy of a software application; encoding the associated license information into the copy of the software application using a watermarking process; providing the copy of the software application having license information encoded therein to a user; and, comparing information received by a user with the encoded license information.

53. (new) The system of claim 52, wherein the encoding is controlled by a key.

54. (new) The system of claim 52, wherein the step of comparing the user supplied information with the encoded license information enables authorization of the software.

55. (new) The system of claim 53, wherein the key is fixed prior to distribution of the software.

56. (new) The system of claim 52, wherein the license information comprises code which affects functionality of the watermarked software.

57. (new) The system of claim 52, wherein the watermark software is resistant to code analysis and/or tampering.

DATA PROTECTION METHOD AND DEVICE

Abstract of the Disclosure

An apparatus and method for encoding and decoding additional information into a digital information in an integral manner. More particularly, the invention relates to a method and device for data protection.

DATA PROTECTION METHOD AND DEVICE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a divisional of U.S. Patent Application Serial No. 10/602,777, [[This application]] which is a continuation application of U.S. Patent Application Serial No. 09/046,627 ([[now awaiting issuance]] which issued July 22, 2003, as U.S. Patent No. 6,598,162), which is a continuation-in-part of U.S. Patent Application Serial No. 08/587,943, filed Jan. 17, 1996, (which issued April 28, 1998, as U.S. Patent No. 5,745,943). The entire disclosure of U.S. Patent Application No. 09/046,627 (which issued July 22, 2003, as U.S. Patent No. 6,598,162) and U.S. Patent Application Serial No. 08/587,943, filed Jan. 17, 1996, (which issued April 28, 1998, as U.S. Patent No. 5,745,943) [[is]] are hereby incorporated by reference in their entireties.

FIELD OF THE INVENTION

[0002] The invention relates to the protection of digital information. More particularly, the invention relates to a method [[for combining transfer functions with predetermined key creation]] and device for data protection.

[0003] With the advent of computer networks and digital multimedia, protection of intellectual property has become a prime concern for creators and publishers of digitized copies of copyrightable works, such as musical recordings, movies, video games, and computer software. One method of protecting copyrights in the digital domain is to use "digital watermarks."

[0004] The prior art includes copy protection systems attempted at many stages in the development of the software industry. These may be various methods by which a software engineer can write the software in a clever manner to determine if it has been copied, and if so to deactivate itself. Also included are undocumented changes to the storage format of the content. Copy protection was generally abandoned by the software industry, since pirates were generally just as clever as the software engineers and figured out ways to modify the software and deactivate the protection. The cost of developing such protection was not justified considering the level of piracy which occurred despite the copy protection.

[0005] Other methods for protection of computer software include the requirement of entering certain numbers or facts that may be included in a packaged software's manual, when prompted at start-up These may be overcome if copies of the manual are distributed to unintended users, or by patching the code to bypass these measures. Other methods include requiring a user to contact the software vendor and to receive "keys" for unlocking software after registration attached to some payment scheme, such as credit card authorization. Further methods include network-based searches of a user's hard drive and comparisons between what is registered to that user and what is actually installed on the user's general computing device. Other proposals, by such parties as AT&T's Bell Laboratories, use "kerning" or actual distance in pixels, in the rendering of text documents, rather than a varied number of ASCII characters. However, this approach can often be defeated by graphics processing analogous to sound processing, which randomizes that information. All of these methods require outside determination and verification of the validity of the software license.

[0006] Digital watermarks can be used to mark each individual copy of a digitized work with information identifying the title, copyright holder, and even the licensed owner of a particular copy. When marked with licensing and ownership information, responsibility is created for individual copies where before there was none. Computer application programs can be watermarked by watermarking digital content resources used in conjunction with images or audio data. Digital watermarks can be encoded with random or pseudo random keys, which act as secret maps for locating the watermarks. These keys make it impossible for a party to find the watermark without having the key. In addition, the encoding method can be enhanced to force a party to cause damage to a watermarked data stream when trying to erase a random-key watermark. Other information is disclosed in "Technology: Digital Commerce", Denise Caruso, New York Times, Aug. 7, 1995; and "Copyrighting in the Information Age", Harley Ungar, ONLINE MARKETPLACE, September 1995, Jupiter Communications.

[0007] Additionally, other methods for hiding information signals in content signals, are disclosed in U.S. Pat. No. 5,319,735--Preuss et al. and U.S. Pat. No. 5,379,345--Greenberg.

[0008] It is desirable to use a "stega-cipher" or watermarking process to hide the necessary parts or resources of the executable object code in the digitized sample resources. It is also desirable to further modify the underlying structure of an executable computer application such that it is more resistant to attempts at patching and analysis by memory capture. A computer application seeks to provide a user with certain utilities or tools, that is, users interact with a computer or similar device to accomplish various tasks and applications provide the relevant interface. Thus, a level of authentication can also be introduced into software, or "digital products," that include digital content, such as audio, video, pictures or multimedia, with digital watermarks. Security is maximized because erasing this code watermark without a key results in the destruction of one or more essential parts of the underlying application, rendering the "program" useless to the unintended user who lacks the appropriate key. Further, if the key is linked to a license code by means of a mathematical function, a mechanism for identifying the licensed owner of an application is created.

[0009] It is also desirable to randomly reorganize program memory structure intermittently during program run time, to prevent attempts at memory capture or object code analysis aimed at eliminating licensing or ownership information, or otherwise modifying, in an unintended manner, the functioning of the application.

[0010] In this way, attempts to capture memory to determine underlying functionality or provide a "patch" to facilitate unauthorized use of the "application," or computer program, without destroying the functionality and thus usefulness of a copyrightable computer program can be made difficult or impossible.

[0011] It is thus the goal of the present invention to provide a higher level of copyright security to object code on par with methods described in digital watermarking systems for digitized media content such as pictures, audio, video and multimedia content in its multifarious forms, as described in previous disclosures, "Steganographic Method and Device" Ser. No. 08/489,172, filed Jun. 7, 1995, now U.S. Pat. No. 5,613,004, and "Human Assisted Random Key Generation and Application for Digital Watermark System", Ser. No. 08/587,944, filed on Jan. 17, 1996, the disclosure of which is hereby incorporated by reference.

[0012] It is a further goal of the present invention to establish methods of copyright protection that can be combined with such schemes as software metering, network distribution of code and specialized protection of software that is designed to work over a network, such as that proposed by Sun Microsystems in their HotJava browser and Java programming language, and manipulation of application code in proposed distribution of documents that can be exchanged with resources or the look and feel of the document being preserved over a network. Such systems are currently being offered by companies including Adobe, with their Acrobat software. This latter goal is accomplished primarily by means of the watermarking of font, or typeface, resources included in applications or documents, which determine how a bitmap representation of the document is ultimately drawn on a presentation device.

[0013] The present invention includes an application of the technology of "digital watermarks." As described in previous disclosures, "Steganographic Method and Device" and "Human Assisted Random Key Generation and Application for Digital Watermark System," watermarks are particularly suitable to the identification, metering, distributing and authenticating digitized content such as pictures, audio, video and derivatives thereof under the description of "multimedia content." Methods have been described for combining both cryptographic methods, and steganography, or hiding something in plain view. Discussions of these technologies can be found in Applied Cryptography by Bruce Schneier and The Code Breakers by David Kahn. For more information on prior art public-key cryptosystems see U.S. Pat. No. 4,200,770 Diffie-Hellman, U.S. Pat. No. 4,218,582 Hellman, U.S. Pat. No. 4,405,829 RSA, U.S. Pat. No. 4,424,414 Hellman Pohlig. Computer code, or machine language instructions, which are not digitized and have zero tolerance for error, must be protected by derivative or alternative methods, such as those disclosed in this invention, which focuses on watermarking with "keys" derived from license codes or other ownership identification information, and using the watermarks encoded with such keys to hide an essential subset of the application code resources.

BACKGROUND OF THE INVENTION

[0014] Increasingly, commercially valuable information is being created and stored in "digital" form. For example, music, photographs and video can all be stored and transmitted as a series of numbers, such as 1's and 0's. Digital techniques let the original information be recreated in a very accurate manner. Unfortunately, digital techniques also let the information be easily copied without the information owner's permission.

[0015] Because unauthorized copying is clearly a disincentive to the digital distribution of valuable information, it is important to establish responsibility for copies and derivative copies of such works. For example, if each authorized digital copy of a popular song is identified with a unique number, any unauthorized copy of the song would also contain the number. This would allow the owner of the information, such as a song publisher, to investigate who made the unauthorized copy. Unfortunately, it is possible that the unique number could be erased or altered if it is simply tacked on at the beginning or end of the digital information.

[0016] As will be described, known digital "watermark" techniques give creators and publishers of digitized multimedia content localized, secured identification and authentication of that content. In considering the various forms of multimedia content, such as "master," stereo, National Television Standards Committee (NTSC) video, audio tape or compact disc, tolerance of quality will vary with individuals and affect the underlying commercial and aesthetic value of the content. For example, if a digital version of a popular song sounds distorted, it will be less valuable to users. It is therefore desirable to embed copyright, ownership or purchaser information, or some combination of these and related data, into the content in a way that will damage the content if the watermark is removed without authorization.

[0017] To achieve these goals, digital watermark systems insert ownership information in a way that causes little or no noticeable effects, or "artifacts," in the underlying content signal. For example, if a digital watermark is inserted into a digital version of a song, it is important that a listener not be bothered by the slight changes introduced by the watermark. It is also important for the watermark technique to maximize the encoding level and "location sensitivity" in the signal to force damage to the content signal when removal is attempted. Digital watermarks address many of these concerns, and research in the field has provided extremely robust and secure implementations.

[0018] What has been overlooked in many applications described in the art, however, are systems which closely mimic distribution of content as it occurs in the real world. For instance, many watermarking systems require the original un-watermarked content signal to enable detection or decode operations. These include highly publicized efforts by NEC, Digimarc and others. Such techniques are problematic because, in the real world, original master copies reside in a rights holders vaults and are not readily available to the public.

[0019] With much activity overly focused on watermark survivability, the security of a digital watermark is suspect. Any simple linear operation for encoding information into a signal may be used to erase the embedded signal by inverting the process. This is not a difficult task, especially when detection software is a plug-in freely available to the public, such as with Digimarc. In general, these systems seek to embed cryptographic information, not cryptographically embed information into target media content.

[0020] Other methods embed ownership information that is plainly visible in the media signal, such as the method described in U.S. Pat. No. 5,530,739 to Braudaway et al. The system described in Braudaway protects a digitized image by encoding a visible watermark to deter piracy. Such an implementation creates an immediate weakness in securing the embedded information because the watermark is plainly visible. Thus, no search for the embedded signal is necessary and the watermark can be more easily removed or altered. For example, while certainly useful to some rights owners, simply placing the symbol "©" in the digital information would only provide limited protection. Removal by adjusting the brightness of the pixels forming the "©" would not be difficult with respect to the computational resources required.

[0021] Other relevant prior art includes U.S. Pat. No. 4,979,210 and 5,073,925 to Nagata et al., which encodes information by modulating an audio signal in the amplitude/time domain. The modulations introduced in the Nagata process carry a "copy/don't copy" message, which is easily found and circumvented by one skilled in the art. The granularity of encoding is fixed by the amplitude and frequency modulation limits required to maintain inaudibility. These limits are relatively low, making it impractical to encode more information using the Nagata process.

[0022] Although U.S. Pat. No. 5,661,018 to Leighton describes a means to prevent collusion attacks in digital watermarks, the disclosed method may not actually provide the security described. For-example, in cases where the watermarking technique is linear, the "insertion envelope" or "watermarking space" is well-defined and thus susceptible to attacks less sophisticated than collusion by unauthorized parties. Over-encoding at the watermarking encoding level is but one simple attack in such linear implementations. Another consideration not made by Leighton is that commercially-valuable content may already exist in a un-watermarked form somewhere, easily accessible to potential pirates, gutting the need for any type of collusive activity. Digitally signing the embedded signal with preprocessing of watermark data is more likely to prevent successful collusion. Furthermore, a "baseline" watermark as disclosed is quite subjective. It is simply described elsewhere in the art as the "perceptually significant" regions of a signal. Making a watermarking function less linear or inverting the insertion of watermarks would seem to provide the same benefit without the additional work required to create a "baseline" watermark. Indeed, watermarking algorithms should already be capable of defining a target insertion envelope or region without additional steps. What is evident is the Leighton patent does not allow for initial prevention of attacks on an embedded watermark as the content is visibly or audibly unchanged.

[0023] It is also important that any method for providing security also function with broadcasting media over networks such as the Internet, which is also referred to as "streaming." Commercial "plug-in" products such as RealAudio and RealVideo, as well as applications by vendors VDONet and Xtreme, are common in such network environments. Most digital watermark implementations focus on common file base signals and fail to anticipate the security of streamed signals. It is desirable that any protection scheme be able to function with a plug-in player without advanced knowledge of the encoded media stream.

[0024] Other technologies focus solely on file-based security. These technologies illustrate the varying applications for security that must be evaluated for different media and distribution environments. Use of cryptolopes or cryptographic containers, as proposed by IBM in its Cryptolope product, and InterTrust, as described in U.S. Pat. Nos. 4,827,508, 4,977,594, 5,050,213 and 5,410,598, may discourage certain forms of piracy. Cryptographic containers, however, require a user to subscribe to particular decryption software to decrypt data. IBM's InfoMarket and InterTrust's DigiBox, among other implementations, provide a generalized model and need proprietary architecture to function. Every user must have a subscription or registration with the party which encrypts the data. Again, as a form of general encryption, the data is scrambled or encrypted without regard to the media and its formatting. Finally, control over copyrights or other neighboring rights is left with the implementing party, in this case, IBM, InterTrust or a similar provider.

[0025] Methods similar to these "trusted systems" exist, and Cerberus Central Limited and Liquid Audio, among a number of companies, offer systems which may functionally be thought of as subsets of IBM and InterTrust's more generalized security offerings. Both Cerberus and Liquid Audio propose proprietary player software which is registered to the user and "locked" in a manner parallel to the locking of content that is distributed via a cryptographic container. The economic trade-off in this model is that users are required to use each respective companies' proprietary player to play or otherwise manipulate content that is downloaded. If, as is the case presently, most music or other media is not available via these proprietary players and more companies propose non-compatible player formats, the proliferation of players will continue. Cerberus and Liquid Audio also by way of extension of their architectures provide for "near-CD quality" but proprietary compression. This requirement stems from the necessity not to allow content that has near-identical data make-up to an existing consumer electronic standard, in Cerberus and Liquid Audio's case the so-called Red Book audio CD standard of 16 bit 44.1 kHz, so that comparisons with the proprietary file may not yield how the player is secured. Knowledge of the player's file format renders its security ineffective as a file may be replicated and played on any common player, not the intended proprietary player of the provider of previously secured and uniquely formatted content. This is the parallel weakness to public key crypto-systems which have gutted security if enough plain text and cipher text comparisons enable a pirate to determine the user's private key.

[0026] Many approaches to digital watermarking leave detection and decoding control with the implementing party of the digital watermark, not the creator of the work to be protected. A set of secure digital watermark implementations address this fundamental control issue forming the basis of key-based approaches. These are covered by the following patents and pending applications, the entire disclosures of which are hereby incorporated by reference: U.S. Pat. No. 5,613, 004 entitled "Steganographic Method and Device" and its derivative U.S. patent application Ser. No. 08/775,216 (which issued November 11, 1997, as U.S. Patent No. 5,687,236), U.S. patent application Ser. No. 08/587,944 entitled "Human Assisted Random Key Generation and Application for Digital Watermark System[[,]]"(which issued October 13, 1998, as U.S. Patent No. 5,822,432), U.S. patent application Ser. No. 08/587,943 entitled "Method for Stega-Cipher Protection of Computer Code[[,]]"(which issued April 28, 1998, as U.S. Patent No. 5,748,569), U.S. patent application Ser. No. 08/677,435 entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data[[,]]"(which issued March 30, 1999, as U.S. Patent No. 5,889,868) and U.S. patent application Ser. No. 08/772,222 entitled "Z-Transform Implementation of Digital Watermarks[[.]]"(which issued June 20, 2000, as U.S. Patent No. 6,078,664). Public key crypto-systems are described in U.S. Pat. No. 4,200,770, 4,218,582, 4,405,829 and 4,424,414, the entire disclosures of which are also hereby incorporated by reference.

[0027] In particular, an improved protection scheme is described in "Method for Stega-Cipher Protection of Computer Code," U.S. patent application Ser. No. 08/587,943 [[,]] (which issued April 28, 1998, as U.S. Patent No. 5,748,569). This technique uses the key-based insertion of binary executable computer code within a content signal that is subsequently, and necessarily, used to play or otherwise manipulate the signal in which it is encoded. With this system, however, certain computational requirements, such as one digital player per digital copy of content, may be necessitated. For instance, a consumer may download many copies of watermarked content. With this technique, the user would also be downloading as many copies of the digital player program. While this form of security may be desirable for some applications, it is not appropriate in many circumstances.

[0028] Finally, even when digital information is distributed in encoded form, it may be desirable to allow unauthorized users to play the information with a digital player, perhaps with a reduced level of quality. For example, a popular song may be encoded and freely distributed in encoded form to the public. The public, perhaps using commonly available plug-in digital players, could play the encoded content and hear the music in some degraded form. The music may-sound choppy, or fuzzy or be degraded in some other way. This lets the public decide, based on the available lower quality version of the song, if they want to purchase a key from the publisher to decode, or "clean-up," the content. Similar approaches could be used to distribute blurry pictures or low quality video. Or even "degraded" text, in the sense that only authenticated portions of the text can be determined with the predetermined key or a validated digital signature for the intended message.

[0029] In view of the foregoing, it can be appreciated that a substantial need exists for a method allowing encoded content to be played, with degraded quality, by a plug-in digital player, and solving the other problems discussed above.

SUMMARY OF THE INVENTION

[0030] The disadvantages of the art are alleviated to a great extent by a method for combining transfer functions with predetermined key creation. In one embodiment, digital information, including a digital sample and format information, is protected by identifying and encoding a portion of the format information. Encoded digital information, including the digital sample and the encoded format information, is generated to protect the original digital information.

[0031] In another embodiment, a digital signal, including digital samples in a file format having an inherent granularity, is protected by creating a predetermined key. The predetermined key is comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

[0032] It is thus a goal of the present invention, to provide a level of security for executable code on similar grounds as that which can be provided for digitized samples. Furthermore, the present invention differs from the prior art in that it does not attempt to stop copying, but rather, determines responsibility for a copy by ensuring that licensing information must be preserved in descendant copies from an original. Without the correct license information, the copy cannot function.

[0033] An improvement over the art is disclosed in the present invention, in that the software itself is a set of commands, compiled by software engineer, which can be configured in such a manner as to tie underlying functionality to the license or authorization of the copy in possession by the user. Without such verification, the functions sought out by the user in the form of software cease to properly work. Attempts to tamper or "patch" substitute code resources can be made highly difficult by randomizing the location of said resources in memory on an intermittent basis to resist most attacks at disabling the system.

[0034] With these and other advantages and features of the invention that will become hereinafter apparent, the nature of the invention may be more clearly understood by reference to the following detailed description of the invention, the appended claims and to the several drawings attached herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0035] FIG. 1 is a block flow diagram of a method for copy protection or authentication of digital information according to an embodiment of the present invention.

DETAILED DESCRIPTION

[0036] In accordance with an embodiment of the present invention, a method combines transfer functions with predetermined key creation. Increased security is achieved in the method by combining elements of "public-key steganography" with cryptographic protocols, which keep in-transit data secure by scrambling the data with "keys" in a manner that is not apparent to those with access to the content to be distributed. Because different forms of randomness are combined to offer robust, distributed security, the present invention addresses an architectural "gray space" between two important areas of security: digital watermarks, a subset of the more general art of steganography, and cryptography. One form of randomness exists in the mask sets that are randomly created to map watermark data into an otherwise unrelated digital signal. The second form of randomness is the random permutations of data formats used with digital players to manipulate the content with the predetermined keys. These forms can be thought of as the transfer function versus the mapping function inherent to digital watermarking processes.

[0037] According to an embodiment of the present invention, a predetermined, or randomly generated, key is used to scramble digital information in a way that is unlike known "digital watermark" techniques and public key crypto-systems. As used herein, a key is also referred to as a "mask set" which includes one or more random or pseudo-random series of bits. Prior to encoding, a mask can be generated by any cryptographically secure random generation process. A block cipher, such as a Data Encryption Standard (DES) algorithm, in combination with a sufficiently random seed value, such as one created using a Message Digest 5 (MD5) algorithm, emulates a cryptographically secure random bit generator. The keys are saved in a database, along with information matching them to the digital signal, for use in descrambling and subsequent viewing or playback. Additional file format or transfer property information is prepared and made available to the encoder, in a bit addressable manner. As well, any authenticating function can be combined, such as Digital Signature Standard (DSS) or Secure Hash Algorithm (SHA).

[0038] Using the predetermined key comprised of a transfer function-based mask set, the data representing the original content is manipulated at the inherent granularity of the file format of the underlying digitized samples. Instead of providing, or otherwise distributing, watermarked content that is not noticeably altered, a partially "scrambled" copy of the content is distributed. The key is necessary both to register the sought-after content and to descramble the content into its original form.

[0039] The present invention uses methods disclosed in "Method for Stega-Cipher Protection of Computer Code," U.S. patent application Ser. No. 08/587,943 (which issued April 28, 1998, as U.S. Patent No. 5,748,569), with respect to transfer functions related to the common file formats, such as PICT, TIFF, AIFF, WAV, etc. Additionally, in cases where the content has not been altered beyond being encoded with such functional data, it is possible for a digital player to still play the content because the file format has not been altered. Thus, the encoded content could still be played by a plug-in digital player as discrete, digitally sampled signals, watermarked or not. That is, the structure of the file can remain basically unchanged by the watermarking process, letting common file format based players work with the "scrambled" content.

[0040] For example, the Compact Disc-Digital Audio (CD-DA) format stores audio information as a series of frames. Each frame contains a number of digital samples representing, for example, music, and a header that contains file format information. As shown in FIG. 1, according to an embodiment of the present invention some of the header information can be identified and "scrambled" using the predetermined key at steps 110 to 130. The music samples can remain unchanged. Using this technique, a traditional CD-DA player will be able to play a distorted version of the music in the sample. The amount of distortion will depend on the way, and extent, that the header, or file format, information has been scrambled. It would also be possible to instead scramble some of the digital samples while leaving the header information alone. In general, the digital signal would be protected by manipulating data at the inherent granularity, or "frames," of the CD-DA file format. To decode the information, a predetermined key is used before playing the digital information at steps 140 and 150.

[0041] A key-based decoder can act as a "plug-in" digital player of broadcast signal streams without foreknowledge of the encoded media stream. Moreover, the data format orientation is used to partially scramble data in transit to prevent unauthorized descrambled access by decoders that lack authorized keys. A distributed key can be used to unscramble the scrambled content because a decoder would understand how to process the key. Similar to on-the-fly decryption operations, the benefits inherent in this embodiment include the fact that the combination of watermarked content security, which is key-based, and the descrambling of the data, can be performed by the same key which can be a plurality of mask sets. The mask sets may include primary, convolution and message delimiter masks with file format data included.

[0042] The creation of an optimized "envelope" for insertion of watermarks provides the basis of much watermark security, but is also a complementary goal of the present invention. The predetermined or random key that is generated is not only an essential map to access the hidden information signal, but is also the descrambler of the previously scrambled signal's format for playback or viewing.

[0043] In a system requiring keys for watermarking content and validating the distribution of the content, different keys may be used to encode different information while secure one way hash functions or one-time pads may be incorporated to secure the embedded signal. The same keys can be used to later validate the embedded digital signature, or even fully decode the digital watermark if desired. Publishers can easily stipulate that content not only be digitally watermarked but that distributors must check the validity of the watermarks by performing digital signature-checks with keys that lack any other functionality. The system can extend to simple authentication of text in other embodiments.

[0044] Before such a market is economically feasible, there are other methods for deploying key-based watermarking coupled with transfer functions to partially scramble the content to be distributed without performing full public key encryption, i.e., a key pair is not necessarily generated, simply, a predetermined key's function is created to re-map the data of the content file in a lossless process. Moreover, the scrambling performed by the present invention may be more dependent on the file in question. Dissimilarly, encryption is not specific to any particular media but is performed on data. The file format remains unchanged, rendering the file useable by any conventional viewer/player, but the signal quality can be intentionally degraded in the absence of the proper player and key. Public-key encryption seeks to completely obscure the sensitive "plaintext" to prevent comparisons with the "ciphertext" to determine a user's private keys. Centralized encryption only differs in the utilization of a single key for both encryption and decryption making the key even more highly vulnerable to attacks to defeat the encryption process. With the present invention, a highly sought after photograph may be hazy to the viewer using any number of commonly available, nonproprietary software or hardware, without the authorized key. Similarly, a commercially valuable song may sound poor.

[0045] The benefit of some form of cryptography is not lost in the present invention. In fact, some piracy can be deterred when the target signal may be known but is clearly being protected through scrambling. What is not anticipated by known techniques, is an ala carte method to change various aspects of file formatting to enable various "scrambled states" for content to be subsequently distributed. An image may lack all red pixels or may not have any of the most significant bits activated. An audio sample can similarly be scrambled to render it less-than-commercially viable.

[0046] The present invention also provides improvements over known network-based methods, such as those used for the streaming of media data over the Internet. By manipulating file formats, the broadcast media, which has been altered to "fit" within electronic distribution parameters, such as bandwidth availability and error correction considerations, can be more effectively utilized to restrict the subsequent use of the content while in transit as well as real-time viewing or playing.

[0047] The mask set providing the transfer function can be read on a per-use basis by issuing an authorized or authenticating "key" for descrambling the signal that is apparent to a viewer or a player or possessor of the authenticating key. The mask set can be read on a per-computer basis by issuing the authorized key that is more generalized for the computer that receives the broadcast signals. Metering and subscription models become viable advantages over known digital watermark systems which assist in designating the ownership of a copy of digitized media content, but do not prevent or restrict the copying or manipulation of the sampled signal in question. For broadcast or streamed media, this is especially the case. Message authentication is also possible, though not guaranteeing the same security as an encrypted file as with general crypto systems.

[0048] The present invention thus benefits from the proprietary player model without relying on proprietary players. No new players will be necessary and existing multimedia file formats can be altered to exact a measure of security which is further increased when coupled with digital watermarks. As with most consumer markets for media content, predominant file formats exist, de facto, and corresponding formats for computers likewise exist. For a commercial compact disc quality audio recording, or 16 bit 44.1 kHz, corresponding file formats include: Audio Interchange File Format (AIFF), Microsoft WAV, Sound Designer II, Sun's .au, Apple's Quicktime, etc. For still image media, formats are similarly abundant: TIFF, PICT, JPEG, GIF, etc. Requiring the use of additional proprietary players, and their complementary file formats, for limited benefits in security is wasteful. Moreover, almost all computers today are multimedia-capable, and this is increasingly so with the popularity of Intel's MMX chip architecture and the PowerPC line of microchips. Because file formatting is fundamental in the playback of the underlying data, the predetermined key can act both as a map, for information to be encoded as watermark data regarding ownership, and a descrambler of the file that has been distributed. Limitations will only exist in how large the key must be retrofitted for a given application, but any manipulation of file format information is not likely to exceed the size of data required versus that for an entire proprietary player.

[0049] As with previous disclosures by the inventor on digital watermarking techniques, the present invention may be implemented with a variety of cryptographic protocols to increase both confidence and security in the underlying system. A predetermined key is described as a set of masks. These masks may include primary, convolution and message delimiter mask. In previous disclosures, the functionality of these masks is defined solely for mapping. The present invention includes a mask set which is also controlled by the distributing party of a copy of a given media signal. This mask set is a transfer function which is limited only by the parameters of the file format in question. To increase the uniqueness or security of each key used to scramble a given media file copy, a secure one way hash function can be used subsequent to transfer properties that are initiated to prevent the forging of a particular key. Public and private keys may be used as key pairs to further increase the unlikeliness that a key may be compromised.

[0050] These same cryptographic protocols can be combined with the embodiments of the present invention in administering streamed content that requires authorized keys to correctly display or play the streamed content in an unscrambled manner. As with digital watermarking, symmetric or asymmetric public key pairs may be used in a variety of implementations. Additionally, the need for certification authorities to maintain authentic key-pairs becomes a consideration for greater security beyond symmetric key implementations. The cryptographic protocols makes possible, as well, a message of text to be authenticated by a message authenticating function in a general computing device that is able to ensure secure message exchanges between authorizing parties.

[0051] An executable computer program is variously referred to as an application, from the point of view of a user, or executable object code from the point of view of the engineer. A collection of smaller, atomic (or indivisible) chunks of object code typically comprise the complete executable object code or application which may also require the presence of certain data resources. These indivisible portions of object code correspond with the programmers' function or procedure implementations in higher level languages, such as C or Pascal. In creating an application, a programmer writes "code" in a higher level language, which is then compiled down into "machine language," or, the executable object code, which can actually be run by a computer, general purpose or otherwise. Each function, or procedure, written in the programming language, represents a self-contained portion of the larger program, and implements, typically, a very small piece of its functionality. The order in which the programmer types the code for the various functions or procedures, and the distribution of and arrangement of these implementations in various files which hold them is unimportant. Within a function or procedure, however, the order of individual language constructs, which correspond to particular machine instructions is important, and so functions or procedures are considered indivisible for purposes of this discussion. That is, once a function or procedure is compiled, the order of the machine instructions which comprise the executable object code of the function is important and their order in the computer memory is of vital importance. Note that many "compilers" perform "optimizations" within functions or procedures, which determine, on a limited scale, if there is a better arrangement for executable instructions which is more efficient than that constructed by the programmer, but does not change the result of the function or procedure. Once these optimizations are performed, however, making random changes to the order of instructions is very likely to "break" the function. When a program is compiled, then, it consists of a collection of these sub-objects, whose exact order or arrangement in memory is not important, so long as any sub-object which uses another sub-object knows where in memory it can be found.

[0052] The memory address of the first instruction in one of these sub-objects is called the "entry point" of the function or procedure. The rest of the instructions comprising that sub-object immediately follow from the entry point. Some systems may prefix information to the entry point which describes calling and return conventions for the code which follows, an example is the Apple Macintosh Operating System (MacOS). These sub-objects can be packaged into what are referred to in certain systems as "code resources," which may be stored separately from the application, or shared with other applications, although not necessarily. Within an application there are also data objects, which consist of some data to be operated on by the executable code. These data objects are not executable. That is, they do not consist of executable instructions. The data objects can be referred to in certain systems as "resources."

[0053] When a user purchases or acquires a computer program, she seeks a computer program that "functions" in a desired manner. Simply, computer software is overwhelmingly purchased for its underlying functionality. In contrast, persons who copy multimedia content, such as pictures, audio and video, do so for the entertainment or commercial value of the content. The difference between the two types of products is that multimedia content is not generally interactive, but is instead passive, and its commercial value relates more on passive not interactive or utility features, such as those required in packaged software, set-top boxes, cellular phones, VCRs, PDAs, and the like. Interactive digital products which include computer code may be mostly interactive but can also contain content to add to the interactive experience of the user or make the underlying utility of the software more aesthetically pleasing. It is a common concern of both of these creators, both of interactive and passive multimedia products, that "digital products" can be easily and perfectly copied and made into unpaid or unauthorized copies. This concern is especially heightened when the underlying product is copyright protected and intended for commercial use.

[0054] The first method of the present invention described involves hiding necessary "parts" or code "resources" in digitized sample resources using a "digital watermarking" process, such as that described in the "Steganographic Method and Device" patent application. The basic premise for this scheme is that there are a certain sub-set of executable code resources, that comprise an application and that are "essential" to the proper function of the application. In general, any code resource can be considered "essential" in that if the program proceeds to a point where it must "call" the code resource and the code resource is not present in memory, or cannot be loaded, then the program fails. However, the present invention uses a definition of "essential" which is more narrow. This is because, those skilled in the art or those with programming experience, may create a derivative program, not unlike the utility provided by the original program, by writing additional or substituted code to work around unavailable resources. This is particularly true with programs that incorporate an optional "plug-in architecture," where several code resources may be made optionally available at run-time. The present invention is also concerned with concentrated efforts by technically skilled people who can analyze executable object code and "patch" it to ignore or bypass certain code resources. Thus, for the present embodiment's purposes, "essential" means that the function which distinguishes this application from any other application depends upon the presence and use of the code resource in question. The best candidates for this type of code resources are NOT optional, or plug-in types, unless special care is taken to prevent work-arounds.

[0055] Given that there are one or more of these essential resources, what is needed to realize the present invention is the presence of certain data resources of a type which are amenable to the "stega-cipher" process described in the "Steganographic Method and Device" patent U.S. Pat. No. 5,613,004. Data which consists of image or audio samples is particularly useful. Because this data consists of digital samples, digital watermarks can be introduced into the samples. What is further meant is that certain applications include image and audio samples which are important to the look and feel of the program or are essential to the processing of the application's functionality when used by the user. These computer programs are familiar to users of computers but also less obvious to users of other devices that run applications that are equivalent in some measure of functionality to general purpose computers including, but not limited to, set-top boxes, cellular phones, "smart televisions," PDAs and the like. However, programs still comprise the underlying "operating systems" of these devices and are becoming more complex with increases in functionality.

[0056] One method of the present invention is now discussed. When code and data resources are compiled and assembled into a precursor of an executable program the next step is to use a *utility* application for final assembly of the executable application. The programmer marks several essential code resources in a list displayed by the utility. The utility will choose one or several essential code resources, and encode them into one or several data resources using the stegacipher process. The end result will be that these essential code resources are not stored in their own partition, but rather stored as encoded information in data resources. They are not accessible at run-time without the key. Basically, the essential code resources that provide functionality in the final end-product, an executable application or computer program, are no longer easily and recognizably available for manipulation by those seeking to remove the underlying copyright or license, or its equivalent information, or those with skill to substitute alternative code resources to "force" the application program to run as an unauthorized copy. For the encoding of the essential code resources, a "key" is needed. Such a key is similar to those described in U.S. Pat. No. 5,613,004, the "Steganographic Method and Device" patent. The purpose of this scheme is to make a particular licensed copy of an application distinguishable from any other. It is not necessary to distinguish every instance of an application, merely every instance of a license. A licensed user may then wish to install multiple copies of an application, legally or with authorization. This method, then, is to choose the key so that it corresponds, is equal to, or is a function of, a license code or license descriptive information, not just a text file, audio clip or identifying piece of information as desired in digital watermarking schemes extant and typically useful to stand-alone, digitally sampled content. The key is necessary to access the underlying code, i.e., what the user understands to be the application program.

[0057] The assembly utility can be supplied with a key generated from a license code generated for the license in question. Alternatively, the key, possibly random, can be stored as a data resource and encrypted with a derivative of the license code. Given the key, it encodes one or several essential resources into one or several data resources. Exactly which code resources are encoded into which data resources may be determined in a random or pseudo random manner. Note further that the application contains a code resource which performs the function of decoding an encoded code resource from a data resource. The application must also contain a data resource which specifies in which data resource a particular code resource is encoded. This data resource is created and added at assembly time by the assembly utility. The application can then operate as follows:

[0058] 1) when it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. This can include a particular computer configuration;

[0059] 2) it stores this information in a personalization data resource;

[0060] 3) Once it has the license code, it can then generate the proper decoding key to access the essential code resources.

[0061] Note that the application can be copied in an uninhibited manner, but must contain the license code issued to the licensed owner, to access its essential code resources. The goal of the invention, copyright protection of computer code and establishment of responsibility for copies, is thus accomplished.

[0062] This invention represents a significant improvement over prior art because of the inherent difference in use of purely informational watermarks versus watermarks which contain executable object code. If the executable object code in a watermark is essential to an application which accesses the data which contains the watermark, this creates an all-or-none situation. Either the user must have the extracted watermark, or the application cannot be used, and hence the user cannot gain full access to the presentation of the information in the watermark bearing data. In order to extract a digital watermark, the user must have a key. The key, in turn, is a function of the license information for the copy of the software in question. The key is fixed prior to final assembly of the application files, and so cannot be changed at the option of the user. That, in turn, means the license information in the software copy must remain fixed, so that the correct key is available to the software. The key and the license information are, in fact, interchangeable. One is merely more readable than the other. In U.S. Pat. No. 5,613,004, the "Steganographic Method and Device, patent", the possibility of randomization erasure attacks on digital watermarks was discussed. Simply, it is always possible to erase a digital watermark, depending on how much damage you are willing to do to the watermark-bearing content stream. The present invention has the significant advantage that you must have the watermark to be able to use the code it contains. If you erase the watermark you have lost a key piece of the functionality of the application, or even the means to access the data which bear the watermark.

[0063] A preferred embodiment would be implemented in an embedded system, with a minimal operating system and memory. No media playing "applets," or smaller sized applications as proposed in new operating environments envisioned by Sun Microsystems and the advent of Sun's Java operating system, would be permanently stored in the system, only the bare necessities to operate the device, download information, decode watermarks and execute the applets contained in them. When an applet is finished executing, it is erased from memory. Such a system would guarantee that content which did not contain readable watermarks could not be used. This is a powerful control mechanism for ensuring that content to be distributed through such a system contains valid watermarks. Thus, in such networks as the Internet or set-top box controlled cable systems, distribution and exchange of content would be made more secure from unauthorized copying to the benefit of copyright holders and other related parties. The system would be enabled to invalidate, by default, any content which has had its watermark(s) erased, since the watermark conveys, in addition to copyright information, the means to fully access, play, record or otherwise manipulate, the content.

[0064] A second method according to the present invention is to randomly re-organize program memory structure to prevent attempts at memory capture or object code analysis. The object of this method is to make it extremely difficult to perform memory capture-based analysis of an executable computer program. This analysis is the basis for a method of attack to defeat the system envisioned by the present invention.

[0065] Once the code resources of a program are loaded into memory, they typically remain in a fixed position, unless the computer operating system finds it necessary to rearrange certain portions of memory during "system time," when the operating system code, not application code, is running. Typically, this is done in low memory systems, to maintain optimal memory utilization. The MacOS for example, uses Handles, which are double-indirect pointers to memory locations, in order to allow the operating system to rearrange memory transparently, underneath a running program. If a computer program contains countermeasures against unlicensed copying, a skilled technician can often take a snapshot of the code in memory, analyze it, determine which instructions comprise the countermeasures, and disable them in the stored application file, by means of a "patch." Other applications for designing code that moves to prevent scanning-tunnelling microscopes, and similar high sensitive hardware for analysis of electronic structure of microchips running code, have been proposed by such parties as Wave Systems. Designs of Wave Systems' microchip are intended for preventing attempts by hackers to "photograph" or otherwise determine "burn in" to microchips for attempts at reverse engineering. The present invention seeks to prevent attempts at understanding the code and its organization for the purpose of patching it. Unlike systems such as Wave Systems', the present invention seeks to move code around in such a manner as to complicate attempts by software engineers to reengineer a means to disable the methods for creating licensed copies on any device that lacks "trusted hardware." Moreover, the present invention concerns itself with any application software that may be used in general computing devices, not chipsets that are used in addition to an underlying computer to perform encryption. Wave Systems' approach to security of software, if interpreted similarly to the present invention, would dictate separate microchip sets for each piece of application software that would be tamperproof. This is not consistent with the economics of software and its distribution.

[0066] Under the present invention, the application contains a special code resource which knows about all the other code resources in memory. During execution time, this special code resource, called a "memory scheduler," can be called periodically, or at random or pseudo random intervals, at which time it intentionally shuffles the other code resources randomly in memory, so that someone trying to analyze snapshots of memory at various intervals cannot be sure if they are looking at the same code or organization from one "break" to the next. This adds significant complexity to their job. The scheduler also randomly relocates itself when it is finished. In order to do this, the scheduler would have to first copy itself to a new location, and then specifically modify the program counter and stack frame, so that it could then jump into the new copy of the scheduler, but return to the correct calling frame. Finally, the scheduler would need to maintain a list of all memory addresses which contain the address of the scheduler, and change them to reflect its new location.

[0067] The methods described above accomplish the purposes of the invention--to make it hard to analyze captured memory containing application executable code in order to create an identifiable computer program or application that is different from other copies and is less susceptible to unauthorized use by those attempting to disable the underlying copyright protection system. Simply, each copy has particular identifying information making that copy different from all other copies.

[0068] Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.

What is claimed is:

1. (original) A method for copy protection of digital information, the digital information including a digital sample and format information, comprising the steps of:

identifying a portion of the format information to be encoded;

generating encoded format information from the identified portion of the format information; and

generating encoded digital information, including the digital sample and the encoded format information.

2. (original) The method of claim 1, further comprising the step of requiring a predetermined key to decode the encoded format information.

3. (original) The method of claim 2, wherein the digital sample and format information are configured to be used with a digital player, and wherein information output from the digital player will have a degraded quality unless the encoded format information is decoded with the predetermined key.

4. (original) The method of claim 3, wherein the information output from the digital player represents a still image, audio or video.

5. (original) The method of claim 3, wherein the information output represents text data to be authenticated.

Claims 6 – 31 (cancelled without prejudice to Applicant's right to seek allowance of said claims in a related application)

32. (new) A method for copy protection of software comprising: embedding the software with a watermark wherein the embedded software operates in a manner substantially the same as the software prior to the embedding step.

33. (new) The process of claim 32, wherein the step of embedding the software with a watermark increases the complexity of code analysis and/or tampering with the software.

34. (new) The process of claim 32, wherein the watermarked software queries a user for personalization information during installation of the software

35. (new) The process of claim 32, wherein the watermark is accessible with a key.

36. (new) The process of claim 35, wherein the key enables authorized use of the watermarked software.

37. (new) The process according to claim 35, wherein the key and license information are interchangeable.

38. (new) The process according to claim 32, wherein the step of embedding the software with a watermark is performed during execution of the software.

39. (new) The process according to claim 32, wherein the step of embedding the software with a watermark modifies the structure of the software being embedded.

40. (new) An article of manufacture comprising a machine readable medium, having thereon stored instructions adapted to be executed by a processor, which instructions when executed result in a process comprising: receiving potentially watermarked software; and identifying the software by extracting the watermark.

41. (new) The article of manufacture of claim 40, wherein the watermark is associated with information fixed prior to distribution of the watermarked software.

42. (new) The article of manufacture of claim 40, wherein the watermark affects functionality of the watermarked software.

43. (new) The article of manufacture of claim 40, wherein the extracted watermark enables generation of a key.

44. (new) The article of manufacture of claim 43, wherein the generated key and licensing information are associated.

45. (new) The article of manufacture of claim 40, further comprising limiting functionality of the software if the watermark cannot be extracted.

46. (new) A method for watermarking software comprising: determining the structure a plurality of code contained in the software; and configuring at least a portion of the plurality of code according to a watermarking process.

47. (new) The process of claim 46, wherein the watermarking process further comprises inserting information into the software after installation.

48. (new) The process of claim 46, wherein the watermarking process configures the at least a portion of the plurality of code according to a key.

49. (new) The process of claim 46, wherein the watermarking process increases the complexity of code analysis and/or tampering with the software.

50. (new) The process of claim 46, wherein the watermarking process is selected from the group comprising: data hiding, steganography or steganographic ciphering.

51. (new) The process of claim 46, wherein the watermarking process is applied during execution of the software.

52. (new) A system for copy protection of software comprising the steps of: associating license information with a copy of a software application; encoding the associated license information into the copy of the software application using a watermarking process; providing the copy of the software application having license information encoded therein to a user; and, comparing information received by a user with the encoded license information.

53. (new) The system of claim 52, wherein the encoding is controlled by a key.

54. (new) The system of claim 52, wherein the step of comparing the user supplied information with the encoded license information enables authorization of the software.

55. (new) The system of claim 53, wherein the key is fixed prior to distribution of the software.

56. (new) The system of claim 52, wherein the license information comprises code which affects functionality of the watermarked software.

57. (new) The system of claim 52, wherein the watermark software is resistant to code analysis and/or tampering.

DATA PROTECTION METHOD AND DEVICE

Abstract of the Disclosure

An apparatus and method for encoding and decoding additional information into a digital information in an integral manner. More particularly, the invention relates to a method and device for data protection.

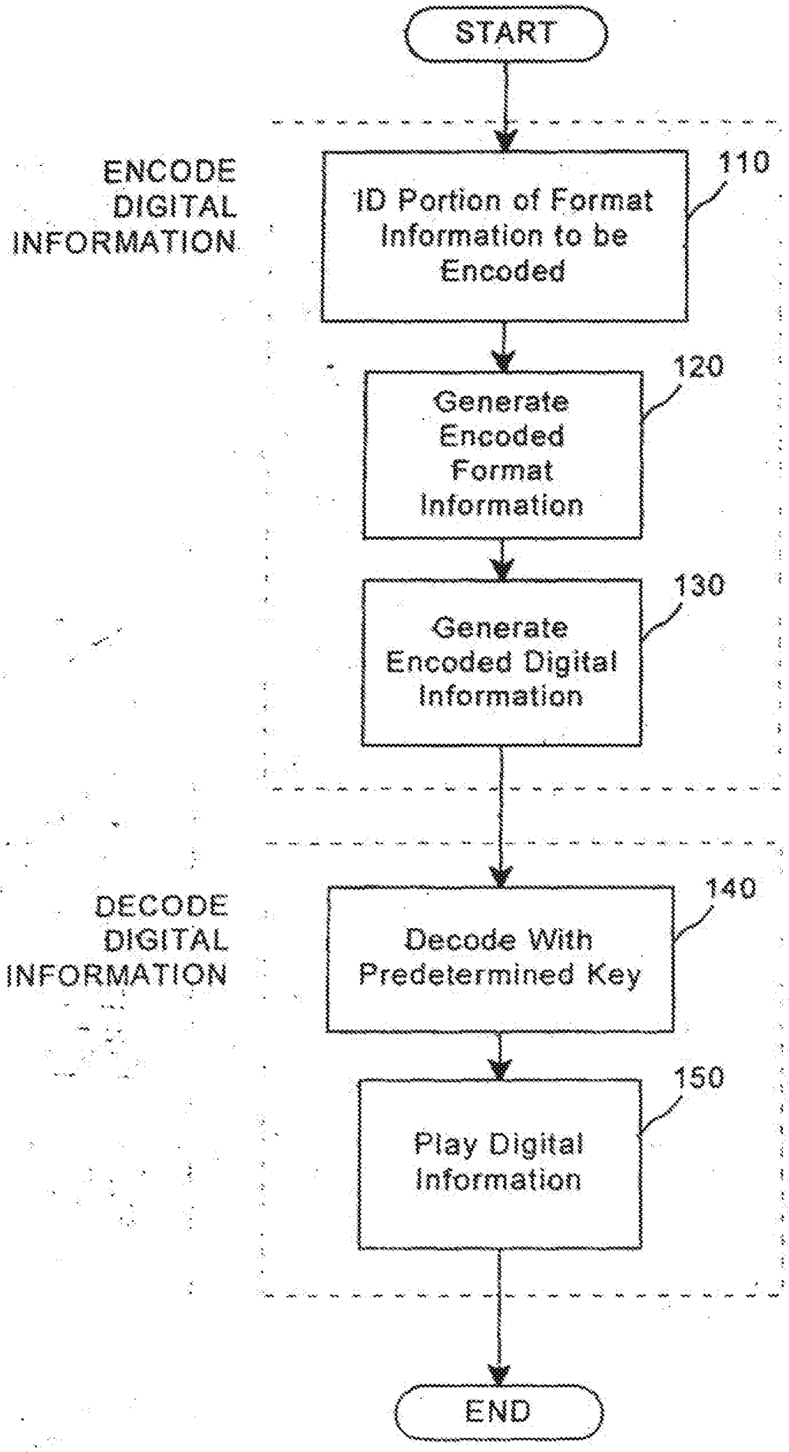

START
ENCODE DIGITAL INFORMATION
ID Portion of Format Information to be Encoded
110
Generate Encoded Format Information
120
Generate Encoded Digital Information
130
DECODE DIGITAL INFORMATION
Decode With Predetermined Key
140
Play Digital Information
150
END

FIG. 1

DECLARATION FOR PATENT APPLICATION

As one of the below named inventors, I hereby declare that:

My residence, post office address and citizenship is as stated below next to my name;

I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

DATA PROTECTION METHOD AND DEVICE

the specification of which: ☒ is attached hereto.
☐ was filed on:
as Application No.:
and was amended on: ____________________.

I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to patentability as defined in 37 C.F.R. § 1.56.

Prior Foreign Application(s)

I hereby claim foreign priority benefits under Title 35, United States Code, § 119(a)-(d) or § 365(b) of any foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed	
				Yes ☐	No ☐
				Yes ☐	No ☐

Prior Provisional Application(s)

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below:

Application Number	Date of Filing (day, month, year)

Prior United States Application(s)

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s), or § 365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Number	Date of Filing (day, month, year)	Status – Patented, Pending, Abandoned
10/602,777	June 25, 2003	Pending
09/046,627	March 24, 1998	Patent No. 6,598,162 July 22, 2003
08/587,943	January 17, 1996	Patent No. 5,745,569 April 28, 1998

All correspondence and telephone communications should be addressed to:

SCOTT MOSKOWITZ
16711 COLLINS AVENUE
NO. 2505
SUNNY ISLES BEACH, FLORIDA 33160

TELEPHONE NUMBER: (305) 956 - 9041
FACSIMILE NUMBER: (305) 956 - 9042

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine and imprisonment, or both, under 18 U.S.C. § 1001, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature ______________________ Date AUGUST 1, 2007

Full Name of First Inventor:	MOSKOWITZ (Family Name)	Scott (First Given Name)	A. (Second Given Name)

Citizenship: United States of America

Residence: 16711 Collins Avenue, # 2505, Sunny Isles Beach, FL 33160

Post Office Address: 16711 Collins Avenue, # 2505, Sunny Isles Beach, FL 33160

APPLICATION DATA SHEET

Application Information

Application Type::	Regular
Subject Matter::	Utility
CD-ROM or CR-R?::	None
Title::	Data Protection Method and Device
Docket No.::	80391.0003CONT2
Request for Early Publication?::	No
Request for Non-Publication?::	No
Total Drawing Sheets::	1
Small Entity::	Yes

Applicant Information

Applicant Authority Type::	First Named Inventor
Primary Citizenship Country::	US
Status::	Full Capacity
Given Name ::	Scott
Middle Name::	A.
Family Name ::	MOSKOWITZ
City of Residence::	Sunny Isles Beach
State of Residence::	FL
Country of Residence::	US
Street of Mailing Address::	16711 Collins Avenue, #2505
City of Mailing Address::	Sunny Isles Beach
State of Mailing Address::	FL
Postal or Zip Code ::	33160

Correspondence Information

Name::	Scott A. Moskowitz
Street of mailing address::	16711 Collins Avenue, #2505
City of mailing address::	Sunny Isles Beach
State of mailing address::	FL
Country of mailing address::	US
Phone Number::	305-956-9041
Facsimile Number ::	305-956-9042
E-Mail Address ::	scott@bluespike.com

Priority Information

Application::	Priority Claim::	Parent Application::	Parent Filing Date::
This Application	Divisional of	10/602,777	06/25/03
10/602,777	Continuation of	09/046,627	03/24/98
09/046,627	Continuation-in-Part of	08/587,943	01/17/96

Initial 08/22/07

UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	Unassigned	Confirmation No. NA
Applicant	:	Scott A. MOSKOWITZ	
Filed	:	Herewith	
TC/A.U.	:	2132	
Examiner	:	Laurel L. LASHLEY	
Docket No.	:	80391.0003CONT2	
Title (before amendment):		Method for Combining Transfer Functions with Predetermined Key Creation	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PRELIMINARY AMENDMENT

Prior to examination on the merits and prior to calculation of the filing fee, please enter the following amendments to the application.

IN THE TITLE:

Please delete the present title and replace it with "DATA PROTECTION METHOD AND DEVICE"

IN THE SPECIFICATION:

On page 1 of the Application, insert the following before the section entitled "Field of the Invention":

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a divisional of U.S. Patent Application Serial No. 10/602,777, which is a continuation application of U.S. Patent Application Serial No. 09/046,627 (which issued July 22, 2003, as U.S. Patent No.

6,598,162), which is a continuation-in-part of U.S. Patent Application Serial No. 08/587,943, filed Jan. 17, 1996, (which issued April 28, 1998, as U.S. Patent No. 5,745,943). The entire disclosure of U.S. Patent Application No. 09/046,627 (which issued July 22, 2003, as U.S. Patent No. 6,598,162) and U.S. Patent Application Serial No. 08/587,943, filed Jan. 17, 1996, (which issued April 28, 1998, as U.S. Patent No. 5,745,943) are hereby incorporated by reference in their entireties.

In the **FIELD OF THE INVENTION**:

After paragraph [0002] please insert the following:

--- With the advent of computer networks and digital multimedia, protection of intellectual property has become a prime concern for creators and publishers of digitized copies of copyrightable works, such as musical recordings, movies, video games, and computer software. One method of protecting copyrights in the digital domain is to use "digital watermarks. "

The prior art includes copy protection systems attempted at many stages in the development of the software industry. These may be various methods by which a software engineer can write the software in a clever manner to determine if it has been copied, and if so to deactivate itself. Also included are undocumented changes to the storage format of the content. Copy protection was generally abandoned by the software industry, since pirates were generally just as clever as the software engineers and figured out ways to modify the software and deactivate the protection. The cost of developing such protection was not justified considering the level of piracy which occurred despite the copy protection.

Other methods for protection of computer software include the requirement of entering certain numbers or facts that may be included in

METHOD FOR STEGA-CIPHER PROTECTION OF COMPUTER CODE

FIELD OF INVENTION

With the advent of computer networks and digital multimedia, protection of intellectual property has become a prime concern for creators and publishers of digitized copies of copyrightable works, such as musical recordings, movies, video games, and computer software. One method of protecting copyrights in the digital domain is to use "digital watermarks."

The prior art includes copy protection systems attempted at many stages in the development of the software industry. These may be various methods by which a software engineer can write the software in a clever manner to determine if it has been copied, and if so to deactivate itself. Also included are undocumented changes to the storage format of the content. Copy protection was generally abandoned by the software industry, since pirates were generally just as clever as the software engineers and figured out ways to modify the software and deactivate the protection. The cost of developing such protection was not justified considering the level of piracy which occurred despite the copy protection.

Other methods for protection of computer software include the requirement of entering certain numbers or facts that may be included in a packaged software's manual, when prompted at start-up. These may be a packaged software's manual, when prompted at start-up. These may be overcome if copies of the manual are distributed to unintended users, or by patching the code to bypass these measures. Other methods include requiring a user to contact the software vendor and to receive "keys" for unlocking software after registration attached to some payment scheme, such as credit card authorization. Further methods include network-based searches of a user's hard drive and comparisons between what is registered to that user and what is actually installed on the user's general computing device. Other proposals, by such parties as AT&T's Bell Laboratories, use "kerning" or actual distance in pixels, in the rendering of text documents, rather than a varied number of ASCII characters. However, this approach can often be defeated by graphics processing analogous to sound processing, which randomizes that information. All of these methods require outside determination and verification of the validity of the software license.

Digital watermarks can be used to mark each individual copy of a digitized work with information identifying the title, copyright holder, and even the licensed owner of a particular copy. When marked with licensing and ownership infomation, responsibility is created for individual copies where before there was none. Computer application programs can be watermarked by watermarking digital content resources used in conjunction with images or audio data. Digital watermarks can be encoded with random or pseudo random keys, which act as secret maps for locating the watermarks. These keys make it impossible for a party to find the watermark without having the key. In addition, the encoding method can be enhanced to force a party to cause damage to a watermarked data stream when trying to erase a random-key watermark. Other information is disclosed in "Technology: Digital Commerce", Denise Caruso, New York Times, Aug. 7, 1995; and "Copyrighting in the Information Age", Harley Ungar, ONLINE MARKETPLACE, September 1995, Jupiter Communications.

Additionally, other methods for hiding information signals in content signals, are disclosed in U.S. Pat. No. 5,319,735--Preuss et al. and U.S. Pat. No. 5,379,345--Greenberg.

It is desirable to use a "stega-cipher" or watermarking process to hide the necessary parts or resources of the executable object code in the digitized sample resources. It is also desirable to further modify the underlying structure of an executable computer application such that it is more resistant to attempts at patching and analysis by memory capture. A computer application seeks to provide a user with certain utilities or tools, that is, users interact with a computer or similar device to accomplish various tasks and applications provide the relevant interface. Thus, a level of authentication can also be introduced into software, or "digital products," that include digital content, such as audio, video, pictures or multimedia, with digital watermarks. Security is maximized because erasing this code watermark without a key results in the destruction of one or more essential parts of the underlying application, rendering the "program" useless to the unintended user who lacks the appropriate key. Further, if the key is linked to a license code by means of a mathematical function, a mechanism for identifying the licensed owner of an application is created.

It is also desirable to randomly reorganize program memory structure intermittently during program run time, to prevent attempts at memory capture or object code analysis aimed at eliminating licensing or ownership information, or otherwise modifying, in an unintended manner, the functioning of the application.

In this way, attempts to capture memory to determine underlying functionality or provide a "patch" to facilitate unauthorized use of the "application," or computer program, without destroying the functionality and thus usefulness of a copyrightable computer program can be made difficult or impossible.

It is thus the goal of the present invention to provide a higher level of copyright security to object code on par with methods described in digital watermarking systems for digitized media content such as pictures, audio, video and multimedia content in its multifarious forms, as described in previous disclosures, "Steganographic Method and Device" Ser. No. 08/489,172, filed Jun. 7, 1995, now U.S. Pat. No. 5,613,004, and "Human Assisted Random Key Generation and Application for Digital Watermark System", Ser. No. 08/587,944, filed on Jan. 17, 1996, the disclosure of which is hereby incorporated by reference.

It is a further goal of the present invention to establish methods of copyright protection that can be combined with such schemes as software metering, network distribution of code and specialized protection of software that is designed to work over a network, such as that proposed by Sun Microsystems in their HotJava browser and Java programming language, and manipulation of application code in proposed distribution of documents that can be exchanged with resources or the look and feel of the document being preserved over a network. Such systems are currently being offered by companies including Adobe, with their Acrobat software. This latter goal is accomplished primarily by means of the watermarking of font, or typeface, resources included in applications or documents, which determine how a bitmap representation of the document is ultimately drawn on a presentation device.

The present invention includes an application of the technology of "digital watermarks." As described in previous disclosures, "Steganographic Method and Device" and "Human Assisted Random Key Generation and Application for Digital Watermark System," watermarks are particularly suitable to the identification, metering, distributing and authenticating digitized content such as pictures, audio, video and derivatives thereof under the description of "multimedia content." Methods have been described for combining both cryptographic methods, and steganography, or hiding something in plain view. Discussions of these technologies can be found in Applied Cryptography by Bruce Schneier and The Code Breakers by David Kahn. For more information on prior art public-key cryptosystems see U.S. Pat. No. 4,200,770 Diffie-Hellman, U.S. Pat. No. 4,218,582 Hellman, U.S. Pat. No. 4,405,829 RSA, U.S. Pat. No. 4,424,414 Hellman Pohlig. Computer code, or machine language instructions, which are not digitized and have zero tolerance for error, must be protected by derivative or alternative methods, such as those disclosed in this invention, which focuses on watermarking with "keys" derived from license codes or other ownership identification information, and using the watermarks encoded with such keys to hide an essential subset of the application code resources. ---

In the **SUMMARY OF THE INVENTION**:

After paragraph [0031] please insert the following:

--- It is thus a goal of the present invention, to provide a level of security for executable code on similar grounds as that which can be provided for digitized samples. Furthermore, the present invention differs from the prior art in that it does not attempt to stop copying, but rather, determines responsibility for a copy by ensuring that licensing information must be preserved in descendant copies from an original. Without the correct license information, the copy cannot function.

An improvement over the art is disclosed in the present invention, in that the software itself is a set of commands, compiled by software engineer, which can be configured in such a manner as to tie underlying functionality to the license or authorization of the copy in possession by the user. Without such verification, the functions sought out by the user in the form of software cease to properly work. Attempts to tamper or "patch" substitute code resources can be made highly difficult by randomizing the location of said resources in memory on an intermittent basis to resist most attacks at disabling the system. ---

In the **DETAILED DESCRIPTION**:

After paragraph [0050] please insert the following:

--- An executable computer program is variously referred to as an application, from the point of view of a user, or executable object code from the point of view of the engineer. A collection of smaller, atomic (or indivisible) chunks of object code typically comprise the complete executable object code or application which may also require the presence of certain data resources. These indivisible portions of object code correspond with the programmers' function or procedure implementations in higher level languages, such as C or Pascal. In creating an application, a programmer writes "code" in a higher level language, which is then compiled down into "machine language," or, the executable object code, which can actually be run by a computer, general purpose or otherwise. Each function, or procedure, written in the programming language, represents a self-contained portion of the larger program, and implements, typically, a very small piece of its functionality. The order in which the programmer types the code for the various functions or procedures, and the distribution of and arrangement of these implementations in various files which hold them is unimportant. Within a function or procedure, however, the order of individual language constructs, which correspond to particular machine instructions is important, and so functions or procedures are considered indivisible for purposes of this discussion. That is, once a function or procedure is compiled, the order of the machine instructions which comprise the executable object code of the function is important and their order in the computer memory is of vital importance. Note that many "compilers" perform "optimizations" within functions or procedures, which determine, on a limited scale, if there is a better arrangement for executable instructions which is more efficient than that constructed by the programmer, but does not change the result of the function or procedure. Once these optimizations are performed, however, making random changes to the order of instructions is very likely to "break" the function. When a program is compiled, then, it consists of a collection of these sub-objects, whose exact order or arrangement in memory is not important, so long as any sub-object which uses another sub-object knows where in memory it can be found.

The memory address of the first instruction in one of these sub-objects is called the "entry point" of the function or procedure. The rest of the instructions comprising that sub-object immediately follow from the entry point. Some systems may prefix information to the entry point which describes calling and return conventions for the code which follows, an example is the Apple Macintosh Operating System (MacOS). These sub-objects can be packaged into what are referred to in certain systems as "code resources," which may be stored separately from the application, or shared with other applications, although not necessarily. Within an application there are also data objects, which consist of some data to be operated on by the executable code. These data objects are not executable. That is, they do not consist of executable instructions. The data objects can be referred to in certain systems as "resources."

When a user purchases or acquires a computer program, she seeks a computer program that "functions" in a desired manner. Simply, computer software is overwhelmingly purchased for its underlying functionality. In contrast, persons who copy multimedia content, such as pictures, audio and video, do so for the entertainment or commercial value of the content. The difference between the two types of products is that multimedia content is not generally interactive, but is instead passive, and its commercial value relates more on passive not interactive or utility features, such as those required in packaged software, set-top boxes, cellular phones, VCRs, PDAs, and the like. Interactive digital products which include computer code may be mostly interactive but can also contain content to add to the interactive experience of the user or make the underlying utility of the software more aesthetically pleasing. It is a common concern of both of these creators, both of interactive and passive multimedia products, that "digital products" can be easily and perfectly copied and made into unpaid or unauthorized copies. This concern is especially heightened when the underlying product is copyright protected and intended for commercial use.

The first method of the present invention described involves hiding necessary "parts" or code "resources" in digitized sample resources using a "digital watermarking" process, such as that described in the "Steganographic Method and Device" patent application. The basic premise for this scheme is that there are a certain sub-set of executable code resources, that comprise an application and that are "essential" to the proper function of the application. In general, any code resource can be considered "essential" in that if the program proceeds to a point where it must "call" the code resource and the code resource is not present in memory, or cannot be loaded, then the program fails. However, the present invention uses a definition of "essential" which is more narrow. This is because, those skilled in the art or those with programming experience, may create a derivative program, not unlike the utility provided by the original program, by writing additional or substituted code to work around unavailable resources. This is particularly true with programs that incorporate an optional "plug-in architecture," where several code resources may be made optionally available at run-time. The present invention is also concerned with concentrated efforts by technically skilled people who can analyze executable object code and "patch" it to ignore or bypass certain code resources. Thus, for the present embodiment's purposes, "essential" means that the function which distinguishes this application from any other application depends upon the presence and use of the code resource in question. The best candidates for this type of code resources are NOT optional, or plug-in types, unless special care is taken to prevent work-arounds.

Given that there are one or more of these essential resources, what is needed to realize the present invention is the presence of certain data resources of a type which are amenable to the "stega-cipher" process described in the "Steganographic Method and Device" patent U.S. Pat. No. 5,613,004. Data which consists of image or audio samples is particularly useful. Because this data consists of digital samples, digital watermarks can be introduced into the samples. What is further meant is that certain applications include image and audio samples which are important to the look and feel of the program or are essential to the processing of the application's functionality when used by the user. These computer programs are familiar to users of computers but also less obvious to users of other devices that run applications that are equivalent in some measure of functionality to general purpose computers including, but not limited to, set-top boxes, cellular phones, "smart televisions," PDAs and the like. However, programs still comprise the underlying "operating systems" of these devices and are becoming more complex with increases in functionality.

One method of the present invention is now discussed. When code and data resources are compiled and assembled into a precursor of an executable program the next step is to use a utility application for final assembly of the executable application. The programmer marks several essential code resources in a list displayed by the utility. The utility will choose one or several essential code resources, and encode them into one or several data resources using the stegacipher process. The end result will be that these essential code resources are not stored in their own partition, but rather stored as encoded information in data resources. They are not accessible at run-time without the key. Basically, the essential code resources that provide functionality in the final end-product, an executable application or computer program, are no longer easily and recognizably available for manipulation by those seeking to remove the underlying copyright or license, or its equivalent information, or those with skill to substitute alternative code resources to "force" the application program to run as an unauthorized copy. For the encoding of the essential code resources, a "key" is needed. Such a key is similar to those described in U.S. Pat. No. 5,613,004, the "Steganographic Method and Device" patent. The purpose of this scheme is to make a particular licensed copy of an application distinguishable from any other. It is not necessary to distinguish every instance of an application, merely every instance of a license. A licensed user may then wish to install multiple copies of an application, legally or with authorization. This method, then, is to choose the key so that it corresponds, is equal to, or is a function of, a license code or license descriptive information, not just a text file, audio clip or identifying piece of information as desired in digital watermarking schemes extant and typically useful to stand-alone, digitally sampled content. The key is necessary to access the underlying code, i.e., what the user understands to be the application program.

The assembly utility can be supplied with a key generated from a license code generated for the license in question. Alternatively, the key, possibly random, can be stored as a data resource and encrypted with a derivative of the license code. Given the key, it encodes one or several essential resources into one or several data resources. Exactly which code resources are encoded into which data resources may be determined in a random or pseudo random manner. Note further that the application contains a code resource which performs the function of decoding an encoded code resource from a data resource. The application must also contain a data resource which specifies in which data resource a particular code resource is encoded. This data resource is created and added at assembly time by the assembly utility. The application can then operate as follows:

1) when it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. This can include a particular computer configuration;

2) it stores this information in a personalization data resource;

3) Once it has the license code, it can then generate the proper decoding key to access the essential code resources.

Note that the application can be copied in an uninhibited manner, but must contain the license code issued to the licensed owner, to access its essential code resources. The goal of the invention, copyright protection of computer code and establishment of responsibility for copies, is thus accomplished.

This invention represents a significant improvement over prior art because of the inherent difference in use of purely informational watermarks versus watermarks which contain executable object code. If the executable object code in a watermark is essential to an application which accesses the data which contains the watermark, this creates an all-or-none situation. Either the user must have the extracted watermark, or the application cannot be used, and hence the user cannot gain full access to the presentation of the information in the watermark bearing data. In order to extract a digital watermark, the user must have a key. The key, in turn, is a function of the license information for the copy of the software in question. The key is fixed prior to final assembly of the application files, and so cannot be changed at the option of the user. That, in turn, means the license information in the software copy must remain fixed, so that the correct key is available to the software. The key and the license information are, in fact, interchangeable. One is merely more readable than the other. In U.S. Pat. No. 5,613,004, the "Steganographic Method and Device, patent", the possibility of randomization erasure attacks on digital watermarks was discussed. Simply, it is always possible to erase a digital watermark, depending on how much damage you are willing to do to the watermark-bearing content stream. The present invention has the significant advantage that you must have the watermark to be able to use the code it contains. If you erase the watermark you have lost a key piece of the functionality of the application, or even the means to access the data which bear the watermark.

A preferred embodiment would be implemented in an embedded system, with a minimal operating system and memory. No media playing "applets," or smaller sized applications as proposed in new operating environments envisioned by Sun Microsystems and the advent of Sun's Java operating system, would be permanently stored in the system, only the bare necessities to operate the device, download information, decode watermarks and execute the applets contained in them. When an applet is finished executing, it is erased from memory. Such a system would guarantee that content which did not contain readable watermarks could not be used. This is a powerful control mechanism for ensuring that content to be distributed through such a system contains valid watermarks. Thus, in such networks as the Internet or set-top box controlled cable systems, distribution and exchange of content would be made more secure from unauthorized copying to the benefit of copyright holders and other related parties. The system would be enabled to invalidate, by default, any content which has had its watermark(s) erased, since the watermark conveys, in addition to copyright information, the means to fully access, play, record or otherwise manipulate, the content.

A second method according to the present invention is to randomly re-organize program memory structure to prevent attempts at memory capture or object code analysis. The object of this method is to make it extremely difficult to perform memory capture-based analysis of an executable computer program. This analysis is the basis for a method of attack to defeat the system envisioned by the present invention.

Once the code resources of a program are loaded into memory, they typically remain in a fixed position, unless the computer operating system finds it necessary to rearrange certain portions of memory during "system time," when the operating system code, not application code, is running. Typically, this is done in low memory systems, to maintain optimal memory utilization. The MacOS for example, uses Handles, which are double-indirect pointers to memory locations, in order to allow the operating system to rearrange memory transparently, underneath a running program. If a computer program contains countermeasures against unlicensed copying, a skilled technician can often take a snapshot of the code in memory, analyze it, determine which instructions comprise the countermeasures, and disable them in the stored application file, by means of a "patch." Other applications for designing code that moves to prevent scanning-tunnelling microscopes, and similar high sensitive hardware for analysis of electronic structure of microchips running code, have been proposed by such parties as Wave Systems. Designs of Wave Systems' microchip are intended for preventing attempts by hackers to "photograph" or otherwise determine "burn in" to microchips for attempts at reverse engineering. The present invention seeks to prevent attempts at understanding the code and its organization for the purpose of patching it. Unlike systems such as Wave Systems', the present invention seeks to move code around in such a manner as to complicate attempts by software engineers to reengineer a means to disable the methods for creating licensed copies on any device that lacks "trusted hardware." Moreover, the present invention concerns itself with any application software that may be used in general computing devices, not chipsets that are used in addition to an underlying computer to perform encryption. Wave Systems' approach to security of software, if interpreted similarly to the present invention, would dictate separate microchip sets for each piece of application software that would be tamperproof. This is not consistent with the economics of software and its distribution.

Under the present invention, the application contains a special code resource which knows about all the other code resources in memory. During execution time, this special code resource, called a "memory scheduler," can be called periodically, or at random or pseudo random intervals, at which time it intentionally shuffles the other code resources randomly in memory, so that someone trying to analyze snapshots of memory at various intervals cannot be sure if they are looking at the same code or organization from one "break" to the next. This adds significant complexity to their job. The scheduler also randomly relocates itself when it is finished. In order to do this, the scheduler would have to first copy itself to a new location, and then specifically modify the program counter and stack frame, so that it could then jump into the new copy of the scheduler, but return to the correct calling frame. Finally, the scheduler would need to maintain a list of all memory addresses which contain the address of the scheduler, and change them to reflect its new location.

The methods described above accomplish the purposes of the invention-- to make it hard to analyze captured memory containing application executable code in order to create an identifiable computer program or application that is different from other copies and is less susceptible to unauthorized use by those attempting to disable the underlying copyright protection system. Simply, each copy has particular identifying information making that copy different from all other copies. --

IN THE CLAIMS:

Please cancel claims 6-31 without prejudice or disclaimer. Claims 6-31 were previously subject to a restriction requirement. Applicant reserves the right to pursue the subject matter of the original claims in this application and in other applications. This listing of claims will replace all prior versions, and listings, of claims in the application. Please add original claims 1 – 5 and new claims 32 - 57 as follows:

1. (original) A method for copy protection of digital information, the digital information including a digital sample and format information, comprising the steps of:

identifying a portion of the format information to be encoded;

generating encoded format information from the identified portion of the format information; and

generating encoded digital information, including the digital sample and the encoded format information.

2. (original) The method of claim 1, further comprising the step of requiring a predetermined key to decode the encoded format information.

3. (original) The method of claim 2, wherein the digital sample and format information are configured to be used with a digital player, and wherein information output from the digital player will have a degraded quality unless the encoded format information is decoded with the predetermined key.

4. (original) The method of claim 3, wherein the information output from the digital player represents a still image, audio or video.

5. (original) The method of claim 3, wherein the information output represents text data to be authenticated.

Claims 6 – 31 (cancelled without prejudice to Applicant's right to seek allowance of said claims in a related application)

32. (new) A method for copy protection of software comprising: embedding the software with a watermark wherein the embedded software operates in a manner substantially the same as the software prior to the embedding step.

33. (new) The process of claim 32, wherein the step of embedding the software with a watermark increases the complexity of code analysis and/or tampering with the software.

34. (new) The process of claim 32, wherein the watermarked software queries a user for personalization information during installation of the software

35. (new) The process of claim 32, wherein the watermark is accessible with a key.

36. (new) The process of claim 35, wherein the key enables authorized use of the watermarked software.

37. (new) The process according to claim 35, wherein the key and license information are interchangeable.

38. (new) The process according to claim 32, wherein the step of embedding the software with a watermark is performed during execution of the software.

39. (new) The process according to claim 32, wherein the step of embedding the software with a watermark modifies the structure of the software being embedded.

40. (new) An article of manufacture comprising a machine readable medium, having thereon stored instructions adapted to be executed by a processor, which instructions when executed result in a process comprising: receiving potentially watermarked software; and identifying the software by extracting the watermark.

41. (new) The article of manufacture of claim 40, wherein the watermark is associated with information fixed prior to distribution of the watermarked software.

42. (new) The article of manufacture of claim 40, wherein the watermark affects functionality of the watermarked software.

43. (new) The article of manufacture of claim 40, wherein the extracted watermark enables generation of a key.

44. (new) The article of manufacture of claim 43, wherein the generated key and licensing information are associated.

45. (new) The article of manufacture of claim 40, further comprising limiting functionality of the software if the watermark cannot be extracted.

46. (new) A method for watermarking software comprising: determining the structure a plurality of code contained in the software; and configuring at least a portion of the plurality of code according to a watermarking process.

47. (new) The process of claim 46, wherein the watermarking process further comprises inserting information into the software after installation.

48. (new) The process of claim 46, wherein the watermarking process configures the at least a portion of the plurality of code according to a key.

49. (new) The process of claim 46, wherein the watermarking process increases the complexity of code analysis and/or tampering with the software.

50. (new) The process of claim 46, wherein the watermarking process is selected from the group comprising: data hiding, steganography or steganographic ciphering.

51. (new) The process of claim 46, wherein the watermarking process is applied during execution of the software.

52. (new) A system for copy protection of software comprising the steps of: associating license information with a copy of a software application; encoding the associated license information into the copy of the software application using a watermarking process; providing the copy of the software application having license information encoded therein to a user; and, comparing information received by a user with the encoded license information.

53. (new) The system of claim 52, wherein the encoding is controlled by a key.

54. (new) The system of claim 52, wherein the step of comparing the user supplied information with the encoded license information enables authorization of the software.

55. (new) The system of claim 53, wherein the key is fixed prior to distribution of the software.

56. (new) The system of claim 52, wherein the license information comprises code which affects functionality of the watermarked software.

57. (new) The system of claim 52, wherein the watermark software is resistant to code analysis and/or tampering.

REMARKS

This is a divisional application of pending U.S. Patent Application No. 10/602,777, filed June 25, 2003. Applicant has bodily incorporated U.S. Patent Application Serial No. 08/587,943, filed January 17, 1996 (which issued as U.S. Patent No. 5,745,569 on April 28, 1998). The '943 application was expressly incorporated by reference into U.S. Patent Application No. 10/602,777 (see Application at page 1). Applicant has changed the title of this divisional application to better describe the bodily incorporated material and the focus of the claims. Applicant has canceled claims 6 - 31 of U.S. Patent Application No. 10/602,777 (without prejudice or disclaimer) and has added original claims 1 - 5 and new claims 32 - 57. Support for new claims 32 - 57 can be found throughout the specification. This amendment does not add any new matter as that term is defined under 37 CFR § 1.118. Accordingly, Applicant respectfully requests entry of this amendment in its entirety.

It is believed that no other fees are required to ensure entry of the amendments and submits that this application is in condition for allowance, and a notice to this effect is earnestly sought.

Respectfully submitted,

Date: August 24, 2007

By:____________________
Scott A. Moskowitz
16711 Collins Avenue, #2505
Sunny Isles Beach, FL 33160
Tel# (305) 956-9041
Fax# (305) 956-9042

overcome if copies of the manual are distributed to unintended users, or by patching the code to bypass these measures. Other methods include requiring a user to contact the software vendor and to receive "keys" for unlocking software after registration attached to some payment scheme, such as credit card authorization. Further methods include network-based searches of a user's hard drive and comparisons between what is registered to that user and what is actually installed on the user's general computing device. Other proposals, by such parties as AT&T's Bell Laboratories, use "kerning" or actual distance in pixels, in the rendering of text documents, rather than a varied number of ASCII characters. However, this approach can often be defeated by graphics processing analogous to sound processing, which randomizes that information. All of these methods require outside determination and verification of the validity of the software license.

Digital watermarks can be used to mark each individual copy of a digitized work with information identifying the title, copyright holder, and even the licensed owner of a particular copy. When marked with licensing and ownership information, responsibility is created for individual copies where before there was none. Computer application programs can be watermarked by watermarking digital content resources used in conjunction with images or audio data. Digital watermarks can be encoded with random or pseudo random keys, which act as secret maps for locating the watermarks. These keys make it impossible for a party to find the watermark without having the key. In addition, the encoding method can be enhanced to force a party to cause damage to a watermarked data stream when trying to erase a random-key watermark. Digital watermarks are described in "Steganographic Method and Device" - The DICE Company, Serial No. 08/489,172, the disclosure of which is hereby incorporated by reference.

Other information is disclosed in "Technology: Digital Commerce", Denise Caruso, New York Times, August 7, 1995; and "Copyrighting in the Information Age", Harley Ungar, ONLINE MARKETPLACE, September 1995, Jupiter Communications.

Additionally, other methods for hiding information signals in content signals, are disclosed in U.S. Patent No. 5,319,735 - Preuss et al. and U.S. Patent No. 5,379,345 - Greenberg.

It is desirable to use a "stega-cipher" or watermarking process to hide the necessary parts or resources of the executable object code in the digitized sample resources. It is also desirable to further modify the underlying structure of an executable computer application such that it is more resistant to attempts at patching and analysis by memory capture. A computer application seeks to provide a user with certain utilities or tools, that is, users interact with a computer or similar device to accomplish various tasks and applications provide the relevant interface. Thus, a level of authentication can also be introduced into software, or "digital products," that include digital content, such as audio, video, pictures or multimedia, with digital watermarks. Security is maximized because erasing this code watermark without a key results in the destruction of one or more essential parts of the underlying application, rendering the "program" useless to the unintended user who lacks the appropriate key. Further, if the key is linked to a license code by means of a mathematical function, a mechanism for identifying the licensed owner of an application is created.

It is also desirable to randomly reorganize program memory structure intermittently during program run time, to prevent attempts at memory capture or object code analysis aimed at eliminating licensing or ownership information, or otherwise modifying, in an unintended manner, the functioning of the application.

In this way, attempts to capture memory to determine underlying functionality or provide a "patch" to facilitate unauthorized use of the "application," or computer program, without destroying the functionality and thus usefulness of a copyrightable computer program can be made difficult or impossible.

It is thus the goal of the present invention to provide a higher level of copyright security to object code on par with methods described in digital watermarking systems for digitized media content such as pictures, audio, video and multimedia content in its multifarious forms, as described in previous disclosures, "Steganographic Method and Device" and "Human Assisted Random Key Generation and Application for Digital Watermark System", filed on even date herewith, the disclosure of which is hereby incorporated by reference.

It is a further goal of the present invention to establish methods of copyright protection that can be combined with such schemes as software metering, network distribution of code and specialized protection of software that is designed to work over a network, such as that proposed by Sun Microsystems in their HotJava browser and Java programming language, and manipulation of application code in proposed distribution of documents that can be exchanged with resources or the look and feel of the document being preserved over a network. Such systems are currently being offered by companies including Adobe, with their Acrobat software. This latter goal is accomplished primarily by means of the watermarking of font, or typeface, resources included in applications or documents, which determine how a bitmap representation of the document is ultimately drawn on a presentation device.

The present invention includes an application of the technology of "digital watermarks." As described in previous disclosures, "Steganographic Method and

Device" and "Human Assisted Random Key Generation and Application for Digital Watermark System," watermarks are particularly suitable to the identification, metering, distributing and authenticating digitized content such as pictures, audio, video and derivatives thereof under the description of "multimedia content." Methods have been described for combining both cryptographic methods, and steganography, or hiding something in plain view. Discussions of these technologies can be found in Applied Cryptography by Bruce Schneier and The Code Breakers by David Kahn. For more information on prior art public-key cryptosystems see US Pat No 4,200,770 Diffie-Hellman, 4,218,582 Hellman, 4,405,829 RSA, 4,424,414 Hellman Pohlig. Computer code, or machine language instructions, which are not digitized and have zero tolerance for error, must be protected by derivative or alternative methods, such as those disclosed in this invention, which focuses on watermarking with "keys" derived from license codes or other ownership identification information, and using the watermarks encoded with such keys to hide an essential subset of the application code resources.

SUMMARY OF THE INVENTION

It is thus a goal of the present invention, to provide a level of security for executable code on similar grounds as that which can be provided for digitized samples. Furthermore, the present invention differs from the prior art in that it does not attempt to stop copying, but rather, determines responsibility for a copy by ensuring that licensing information must be preserved in descendant copies from an original. Without the correct license information, the copy cannot function.

An improvement over the art is disclosed in the present invention, in that the software itself is a set of commands, compiled by software engineer, which can be configured in such a manner as to tie underlying functionality to the license or authorization of the copy in possession by the user. Without such verification, the functions sought out by the user in the form of software cease to properly work. Attempts to tamper or "patch" substitute code resources can be made highly difficult by randomizing the location of said resources in memory on an intermittent basis to resist most attacks at disabling the system.

DETAILED DESCRIPTION

An executable computer program is variously referred to as an application, from the point of view of a user, or executable object code from the point of view of the engineer. A collection of smaller, atomic (or indivisible) chunks of object code typically comprise the complete executable object code or application which may also require the presence of certain data resources. These indivisible portions of object code correspond with the programmers' function or procedure implementations in higher level languages, such as C or Pascal. In creating an application, a programmer writes "code" in a higher level language, which is then compiled down into "machine language," or, the executable object code, which can actually be run by a computer, general purpose or otherwise. Each function, or procedure, written in the programming language, represents a self-contained portion of the larger program, and implements, typically, a very small piece of its functionality. The order in which the programmer types the code for the various functions or procedures, and the distribution of and arrangement of these implementations in various files which hold them is unimportant. Within a function or procedure, however, the order of individual language constructs, which correspond to particular machine instructions is important, and so functions or procedures are considered indivisible for purposes of this discussion. That is, once a function or procedure is compiled, the order of the machine instructions which comprise the executable object code of the function is important and their order in the computer memory is of vital importance. Note that many "compilers" perform "optimizations" within functions or procedures, which determine, on a limited scale, if there is a better arrangement for executable instructions which is more efficient than that constructed by the programmer, but does not change the result of the function or procedure. Once these optimizations are performed, however, making random changes to the order of instructions is very likely to "break" the function. When a program is compiled, then, it consists of a collection of these sub-objects, whose exact order or arrangement in memory is not important, so long as any sub-object which uses another sub-object knows where in memory it can be found.

The memory address of the first instruction in one of these sub-objects is called the "entry point" of the function or procedure. The rest of the instructions comprising that sub-object immediately follow from the entry point. Some systems may prefix information to the entry point which describes calling and return conventions for the code which follows, an example is the Apple Macintosh Operating System (MacOS). These sub-objects can be packaged into what are referred to in certain systems as "code resources," which may be stored separately from the application, or shared with other applications, although not necessarily. Within an application there are also data objects, which consist of some data to be operated on by the executable code. These data objects are not executable. That is, they do not consist of executable instructions. The data objects can be referred to in certain systems as "resources."

When a user purchases or acquires a computer program, she seeks a computer program that "functions" in a desired manner. Simply, computer software is overwhelmingly purchased for its underlying functionality. In contrast, persons who copy multimedia content, such as pictures, audio and video, do so for the entertainment or commercial value of the content. The difference between the two types of products is that multimedia content is not generally interactive, but is instead passive, and its commercial value relates more on passive not interactive or utility features, such as those required in packaged software, set-top boxes, cellular phones, VCRs, PDAs, and the like. Interactive digital products which include computer code may be mostly interactive but can also contain content to add to the interactive experience of the user or make the underlying utility of the software more aesthetically pleasing. It is a common concern of both of these creators, both of interactive and passive multimedia products, that "digital products" can be easily and perfectly copied and made into unpaid or unauthorized copies. This concern is especially heightened when the underlying product is copyright protected and intended for commercial use.

The first method of the present invention described involves hiding necessary "parts" or code "resources" in digitized sample resources using a "digital watermarking" process, such as that described in the "Steganographic Method and Device" patent application. The basic premise for this scheme is that there are a certain sub-set of executable code resources, that comprise an application and that are "essential" to the proper function of the application. In general, any code resource can be considered "essential" in that if the program proceeds to a point where it must "call" the code resource and the code resource is not present in memory, or cannot be loaded, then the program fails.

However, the present invention uses a definition of "essential" which is more narrow. This is because, those skilled in the art or those with programming experience, may create a derivative program, not unlike the utility provided by the original program, by writing additional or substituted code to work around unavailable resources. This is particularly true with programs that incorporate an optional "plug-in architecture," where several code resources may be made optionally available at run-time. The present invention is also concerned with concentrated efforts by technically skilled people who can analyze executable object code and "patch" it to ignore or bypass certain code resources. Thus, for the present embodiment's purposes, "essential" means that the function which distinguishes this application from any other application depends upon the presence and use of the code resource in question. The best candidates for this type of code resources are NOT optional, or plug-in types, unless special care is taken to prevent work-a-rounds.

Given that there are one or more of these essential resources, what is needed to realize the present invention is the presence of certain data resources of a type which are amenable to the "stega-cipher" process described in the "Steganographic Method and Device" patent application. Data which consists of image or audio samples is particularly useful. Because this data consists of digital samples, digital watermarks can be introduced into the samples. What is further meant is that certain applications include image and audio samples which are important to the look and feel of the program or are essential to the processing of the application's functionality when used by the user. These computer programs are familiar to users of computers but also less obvious to users of other devices that run applications that are equivalent in some measure of functionality to general purpose computers including, but not limited to, set-top boxes, cellular phones, "smart televisions," PDAs and the like. However, programs still comprise the underlying "operating systems" of these devices and are becoming more complex with increases in functionality.

One method of the present invention is now discussed. When code and data resources are compiled and assembled into a precursor of an executable program the next step is to use a utility application for final assembly of the executable application. The programmer marks several essential code resources in a list displayed by the utility. The utility will choose one or several essential code resources, and encode them into one or several data resources using the stega-cipher process. The end result will be that these essential code resources are not stored in their own partition, but rather stored as encoded information in data resources. They are not accessible at run-time without the key. Basically, the essential code resources that provide functionality in the final end-product, an executable application or computer program, are no longer easily and recognizably available for manipulation by those seeking to remove the underlying copyright or license, or its equivalent information, or those with skill to substitute alternative code resources to "force" the application program to run as an unauthorized copy. For the encoding of the essential code resources, a "key" is needed. Such a key is similar to those described in the "Steganographic Method and Device." The purpose of this scheme is to make a particular licensed copy of an application distinguishable from any other. It is not necessary to distinguish every instance of an application, merely every instance of a license. A licensed user may then wish to install multiple copies of an application, legally or with authorization. This method, then, is to choose the key so that it corresponds, is equal to, or is a function of, a license code or license descriptive information, not just a text file, audio clip or identifying piece of information as desired in digital watermarking schemes extant and typically useful to stand-alone, digitally sampled content. The key is necessary to access the underlying code, i.e., what the user understands to be the application program.

The assembly utility can be supplied with a key generated from a license code generated for the license in question. Alternatively, the key, possibly random, can be stored as a data resource and encrypted with a derivative of the license code. Given the key, it encodes one or several essential resources into one or several data resources. Exactly which code resources are encoded into which data resources may be determined in a random or pseudo random manner. Note further that the application contains a code resource which performs the function of decoding an encoded code resource from a data resource. The application must also contain a data resource which specifies in which data resource a particular code resource is encoded. This data resource is created and added at assembly time by the assembly utility. The application can then operate as follows:

1) when it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. This can include a particular computer configuration;

2) it stores this information in a personalization data resource;

3) Once it has the license code, it can then generate the proper decoding key to access the essential code resources.

Note that the application can be copied in an uninhibited manner, but must contain the license code issued to the licensed owner, to access its essential code resources. The goal of the invention, copyright protection of computer code and establishment of responsibility for copies, is thus accomplished.

This invention represents a significant improvement over prior art because of the inherent difference in use of purely informational watermarks versus watermarks which contain executable object code. If the executable object code in a watermark is essential to an application which accesses the data which contains the watermark, this creates an all-or-none situation. Either the user must have the extracted watermark, or the application cannot be used, and hence the user cannot gain full access to the presentation of the information in the watermark bearing data. In order to extract a digital watermark, the user must have a key. The key, in turn, is a function of the license information for the copy of the software in question. The key is fixed prior to final assembly of the application files, and so cannot be changed at the option of the user. That, in turn, means the license information in the software copy must remain fixed, so that the correct key is available to the software. The key and the license information are, in fact, interchangeable. One is merely more readable than the other. In the earlier developed "Steganographic Method and Device," the possibility of randomization erasure attacks on digital watermarks was discussed. Simply, it is always possible to erase a digital watermark, depending on how much damage you are willing to do to the watermark-bearing content stream. The present invention has the significant advantage that you must have the watermark to be able to use the code it contains. If you erase the watermark you have lost a key piece of the functionality of the application, or even the means to access the data which bear the watermark.

A preferred embodiment would be implemented in an embedded system, with a minimal operating system and memory. No media playing "applets," or smaller sized applications as proposed in new operating environments envisioned by Sun Microsystems and the advent of Sun's Java operating system, would be permanently stored in the system, only the bare necessities to operate the device, download information, decode watermarks and execute the applets contained in them. When an applet is finished executing, it is erased from memory. Such a system would guarantee that content which did not contain readable watermarks could not be used. This is a powerful control mechanism for ensuring that content to be distributed through such a system contains valid watermarks. Thus, in such networks as the Internet or set-top box controlled cable systems, distribution and exchange of content would be made more secure from unauthorized copying to the benefit of copyright holders and other related parties. The system would be enabled to invalidate, by default, any content which has had its watermark(s) erased, since the watermark conveys, in addition to copyright information, the means to fully access, play, record or otherwise manipulate, the content.

A second method according to the present invention is to randomly re-organize program memory structure to prevent attempts at memory capture or object code analysis. The object of this method is to make it extremely difficult to perform memory capture-based analysis of an executable computer program. This analysis is the basis for a method of attack to defeat the system envisioned by the present invention.

Once the code resources of a program are loaded into memory, they typically remain in a fixed position, unless the computer operating system finds it necessary to rearrange certain portions of memory during "system time," when the operating system code, not application code, is running. Typically, this is done in low memory systems, to maintain optimal memory utilization. The

MacOS for example, uses Handles, which are double-indirect pointers to memory locations, in order to allow the operating system to rearrange memory transparently, underneath a running program. If a computer program contains countermeasures against unlicensed copying, a skilled technician can often take a snapshot of the code in memory, analyze it, determine which instructions comprise the countermeasures, and disable them in the stored application file, by means of a "patch." Other applications for designing code that moves to prevent scanning-tunnelling microscopes, and similar high sensitive hardware for analysis of electronic structure of microchips running code, have been proposed by such parties as Wave Systems. Designs of Wave Systems' microchip are intended for preventing attempts by hackers to "photograph" or otherwise determine "burn in" to microchips for attempts at reverse engineering. The present invention seeks to prevent attempts at understanding the code and its organization for the purpose of patching it. Unlike systems such as Wave Systems', the present invention seeks to move code around in such a manner as to complicate attempts by software engineers to reengineer a means to disable the methods for creating licensed copies on any device that lacks "trusted hardware." Moreover, the present invention concerns itself with any application software that may be used in general computing devices, not chipsets that are used in addition to an underlying computer to perform encryption. Wave Systems' approach to security of software, if interpreted similarly to the present invention, would dictate separate microchip sets for each piece of application software that would be tamperproof. This is not consistent with the economics of software and its distribution.

Under the present invention, the application contains a special code resource which knows about all the other code resources in memory. During execution time, this special code resource, called a "memory scheduler," can be called periodically, or at random or pseudo random intervals, at which time it intentionally shuffles the other code resources randomly in memory, so that someone trying to analyze snapshots of memory at various intervals cannot be sure if they are looking at the same code or organization from one "break" to the next. This adds significant complexity to their job. The scheduler also randomly relocates itself when it is finished. In order to do this, the scheduler would have to first copy itself to a new location, and then specifically modify the program counter and stack frame, so that it could then jump into the new copy of the scheduler, but return to the correct calling frame. Finally, the scheduler would need to maintain a list of all memory addresses which contain the address of the scheduler, and change them to reflect its new location.

The methods described above accomplish the purposes of the invention - to make it hard to analyze captured memory containing application executable code in order to create an identifiable computer program or application that is different from other copies and is less susceptible to unauthorized use by those attempting to disable the underlying copyright protection system. Simply, each copy has particular identifying information making that copy different from all other copies.

What is Claimed Is:

1. A method of associating executable object code with a digital sample stream by means of a digital watermark wherein the digital watermark contains executable object code and is encoded into the digital sample stream.

2. The method of claim 1 wherein a key to access the digital watermark is a function of a collection of license information pertaining to the software which is accessing the watermark

where license information consists of one or more of the following items:

Owning Organization name;

Personal Owner name;

Owner Address;

License code;

Software serialization number;

Distribution parameters;

Appropriate executable general computing device architecture;

Pricing; and

Software Metering details.

3. The method of claim 1 further comprising the step of transmitting the digital sample stream, via a transmission means, from a publisher to a subscriber

wherein transmission means can selected from the group of

soft sector magnetic disk media;

hard sector magnetic disk media;

magnetic tape media;

optical disc media;

Digital Video Disk media;

magneto-optical disk media;

memory cartridge;

telephone lines;

SCSI;
Ethernet or Token Ring Network;
ISDN;
ATM network;
TCP/IP network;
analog cellular network;
digital cellular network;
wireless network;
digital satellite;
cable network;
fiber optic network; and
electric powerline network.

4. The method of claim 1 where the object code to be encoded is comprised of series of executable machine instructions which perform the function of

processing a digital sample stream for the purpose of modifying it or playing the digital sample stream.

5. The method of claim 3 further comprising the steps of:

decoding said digital watermark and extracting object code;

loading object code into computer memory for the purpose of execution;

executing said object code in order to process said digital sample stream for the purpose of playback.

6. A method of assembling an application to be protected by watermark encoding of essential resources comprising the steps of:

assembling a list of identifiers of essential code resources of an application where identifiers allow the code resource to be accessed and loaded into memory;

providing license information on the licensee who is to receive an individualized copy of the application;

storing license information in a personalization resource which is added to the list of application data resources;

generating a digital watermark key from the license information; using the key as a pseudo-random number string to select a list of suitable digital sample data resources, the list of essential code resources, and a mapping of which essential code resources are to be watermarked into which data resources;

storing the map, which is a list of paired code and data resource identifiers, as a data resource, which is added to the application;

adding a digital watermark decoder code resource to the application, to provide a means for extracting essential code resource from data resources, according to the map;

processing the map list and encoding essential code resources into digital sample data resources with a digital watermark encoder;

removing self-contained copies of the essential code resources which have been watermarked into data resources; and

combining all remaining code and data resources into a single application or installer.

7. A method of intermittently relocating application code resources in computer memory, in order to prevent, discourage, or complicate attempts at memory capture based code analysis.

8. The method of claim 7 additionally comprising the step of

assembling a list of identifiers of code resources of an application where identifiers allow the code resource to be accessed and loaded into memory.

9. The method of claim 8 additionally comprising the step of modifying application program structure to make all code resource calls indirectly, through the memory scheduler, which looks up code resources in its list and dispatches calls.

10. The method of claim 9 additionally comprising the step of intermittently rescheduling or shuffling all code resources prior to or following the dispatch of a code resource call through the memory scheduler.

11. The method of claim 10 additionally comprised of the step of the memory scheduler copying itself to a new location in memory.

12. The method of claim 11 additionally comprising the step of modifying the stack frame, program counter, and memory registers of the CPU to cause the scheduler to jump to the next instruction comprising the scheduler, in the copy, to erase the previous memory instance of the scheduler, and changing all memory references to the scheduler to reflect its new location, and to return from the copy of the scheduler to the frame which called the previous copy of the scheduler.

ABSTRACT OF THE DISCLOSURE:

A method for protecting computer code copyrights by encoding the code into a data resource with a digital watermark. The digital watermark contains licensing information interwoven with essential code resources encoded into data resources. The result is that while an application program can be copied in an uninhibited manner, only the licensed user having the license code can access essential code resources to operate the program and any descendant copies bear the required license code.

PATENT APPLICATION SERIAL NO.________________________

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

```
08/27/2007 HMARZI1  00000069 11695388

01 FC:2011                    150.00 OP
02 FC:2111                    250.00 OP
03 FC:2311                    100.00 OP
04 FC:2201                    200.00 OP
05 FC:2202                    275.00 OP
```

PTO-1556
(5/87)

*U.S. Government Printing Office: 2002- 489-267/69033

Attachment 15 Page 102 of 104

PTO/SB/06 (02-07)
Approved for use through 02/28/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number 11895388

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)		RATE ($)	FEE ($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	$150		N/A	$300
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	$250		N/A	$500
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	$100		N/A	$200
TOTAL CLAIMS (37 CFR 1.16(i))	31 minus 20 =	* 11	x $25 =	275	OR	x $50 =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	5 minus 3 =	* 2	x $100 =	200		x $200 =	
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).		$125			$250	
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))			+180 =			+360 =	
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	975		TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT A

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	x $25 =		OR	x $50 =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	x $100 =		OR	x $200 =	
Application Size Fee (37 CFR 1.16(s))					$125			$250	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					+180 =		OR	+360 =	
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

AMENDMENT B

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	x $25 =		OR	x $50 =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	x $100 =		OR	x $200 =	
Application Size Fee (37 CFR 1.16(s))					$125			$250	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					+180 =		OR	+360 =	
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

CLAIMS ONLY	Application Number 11495388	Filing Date
	Applicant(s)	

CLAIMS

	AS FILED		AFTER FIRST AMENDMENT		AFTER SECOND AMENDMENT	
	IND	DEP	IND	DEP	IND	DEP
1	1					
2		1				
3		1				
4		1				
5		1				
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						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46	1					
47						
48						
49						
50						
Total Indep						
Total Depend						
Total Claims						

	*		*		*	
	IND	DEP	IND	DEP	IND	DEP
51		1				
52	1					
53		1				
54		1				
55		1				
56		1				
57		1				
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93						
94						
95						
96						
97						
98						
99						
100						
Total Indep						
Total Depend						
Total Claims						

NEIFELD REF: SCOT0014-6
CLIENT REF: SCOT0014-6
Application/Patent No: Unknown
USPTO CONF. NO: Unknown
File/Issue Date: Filed Herewith
Inventor: SCOTT MOSKOWITZ
Title: Data protection method and device
Examiner/ArtUnit: Unknown
ENTITY STATUS: Unknown
Priority claims and PCT Intl data: This application is a continuation of U.S. Application No. 11/895,388, filed August 24, 2007, which is a division of U.S. patent application No. 10/602,777, filed June 25, 2003, issued February 16, 2010 as U.S. Patent No. 7,664,263, which is a continuation of U.S. patent application No. 09/046,627, filed March 24, 1998, issued Jul. 22, 2003, as U.S. Pat. No. 6,598,162, which is a continuation-in-part of U.S. patent application No. 08/587,943, filed Jan. 17, 1996, which issued Apr. 28, 1998, as U.S. Pat. No. 5,745,943. The entire disclosure of U.S. Application No. 11/895,388, filed August 24, 2007, U.S. patent application No. 09/046,627 which issued Jul. 22, 2003, as U.S. Pat. No. 6,598,162 and U.S. patent application No. 08/587,943, filed Jan. 17, 1996, which issued Apr. 28, 1998, as U.S. Pat. No. 5,745,943 are hereby incorporated by reference in their entireties.

37 CFR 1.7(c) FILING RECEIPT AND TRANSMITTAL LETTER WITH AUTHORIZATION TO CHARGE DEPOSIT ACCOUNT

1. **FOR 35 USC 371 NATIONAL STAGE FILINGS, ONLY, THE COMMISSIONER IS HEREBY AUTHORIZED TO CHARGE ANY FEES WHICH MAY BE REQUIRED, OR CREDIT ANY OVERPAYMENT, TO DEPOSIT ACCOUNT NUMBER 50-2106.**

2. **FEES (PAID HEREWITH BY EFS CREDIT CARD SUBMISSION) $:**1,250

NEW APPLICATION FILING FEES
1011/2011 1.16(a)(1) Basic filing fee - Utility $380
1111/2111 1.16(k) Utility Search Fee $620
1311/2311 1.16(o) Utility Examination Fee $250

3. **THE FOLLOWING DOCUMENTS ARE SUBMITTED HEREWITH:**
NEW APPLICATION DOCUMENTS

37 CFR 1.115 PRELIMINARY AMENDMENT (4 pages)
SPECIFICATION (22 pages)
CLAIMS (8 pages)
ABSTRACT (1 page)
FIGURES (1 page)
DECLARATION filed in parent application No. 11/895,388, filed August 24, 2007 (3 pages)

4. **FOR INTERNAL NEIFELD IP LAW, PC USE ONLY**

Disbursements: PClaw BankAcct, G/L: **6, 5010**
PCLAW BILLING REFERENCE:SCOT0001
Check#, Entry date, Amount: 1496, 7/22/2012, 1,250

Service Fees: Amount/CreditAtty/Entry date/Services: 400/BTM/7-22-2012/firm charge for paying a gov. fee for application filing

INITIALS OF PERSON WHO ***ENTERED*** ACCOUNTING DATA:
AUTHORIZING SIGNER ON DEPOSIT ACCOUNT:
DATE: 7/24/2012 **SIGNATURE**: /BruceMargulies/
Printed: July 24, 2012 (10:51am) Bruce Margulies, Reg. No. 64,175
Y:\Clients\SCOT Scott A Moskowitz and Wistaria Trading, Inc\SCOT0014-6\Drafts\ApplicationFiling_SCOT0014-6_7-22-2012a.wpd

NEIFELD REF: SCOT0014-6
CLIENT REF: SCOT0014-6
Application/Patent No: Unknown
USPTO CONF. NO: Unknown
File/Issue Date: Filed Herewith
Inventor: SCOTT MOSKOWITZ
Title: Data protection method and device
Examiner/ArtUnit: Unknown
ENTITY STATUS: Unknown
Priority claims and PCT Intl data: This application is a continuation of U.S. Application No. 11/895,388, filed August 24, 2007, which is a division of U.S. patent application No. 10/602,777, filed June 25, 2003, issued February 16, 2010 as U.S. Patent No. 7,664,263, which is a continuation of U.S. patent application No. 09/046,627, filed March 24, 1998, issued Jul. 22, 2003, as U.S. Pat. No. 6,598,162, which is a continuation-in-part of U.S. patent application No. 08/587,943, filed Jan. 17, 1996, which issued Apr. 28, 1998, as U.S. Pat. No. 5,745,943. The entire disclosure of U.S. Application No. 11/895,388, filed August 24, 2007, U.S. patent application No. 09/046,627 which issued Jul. 22, 2003, as U.S. Pat. No. 6,598,162 and U.S. patent application No. 08/587,943, filed Jan. 17, 1996, which issued Apr. 28, 1998, as U.S. Pat. No. 5,745,943 are hereby incorporated by reference in their entireties.

37 CFR 1.115 PRELIMINARY AMENDMENT

ASSISTANT COMMISSIONER FOR PATENTS
ALEXANDRIA, VA 22313

Sir:

Prior to examination on the merits, please amend this application as follows.

I. IN THE SPECIFICATION

At Page 1, please replace Paragraph [0001] with the following paragraph:

[0001] This application is a continuation of U.S. Application No. 11/895,388, filed August 24, 2007, which is a division[[al]] of U.S. patent application [[Ser.]] No. 10/602,777, filed June 25, 2003, issued February 16, 2010 as U.S. Patent No. 7,664,263, which is a continuation ~~application~~ of U.S. patent application [[Ser. No.]] 09/046,627, filed March 24, 1998, ~~(which~~ issued Jul. 22, 2003, as U.S. Pat. No. 6,598,162[[)]], which is a continuation-in-part of U.S. patent application [[Ser.]] No. 08/587,943, filed Jan. 17, 1996, ~~(which~~ issued Apr. 28, 1998, as U.S. Pat. No. 5,745,943[[)]]. The entire disclosure of U.S. Application No. 11/895,388, filed August 24, 2007, U.S. patent application [[Ser.]] No. 09/046,627 ~~(which~~ issued Jul. 22, 2003, as U.S. Pat. No. 6,598,162[[)]] and U.S. patent application [[Ser.]] No. 08/587,943, filed Jan. 17, 1996, ~~(which~~ issued Apr. 28, 1998, as U.S. Pat. No. 5,745,943[[)]] are hereby incorporated by reference in their entireties.

II. IN THE CLAIMS

1. (Original) A method for copy protection of digital information, the digital information including a digital sample and format information, comprising the steps of: identifying a portion of the format information to be encoded; generating encoded format information from the identified portion of the format information; and generating encoded digital information, including the digital sample and the encoded format information.

2-57 (Canceled).

III. REMARKS

This preliminary amendment updates the priority claim and cancels claims 2-57. Claim 1 is pending. The applicant intends to add claims for examination prior to examination.

	Respectfully Submitted,
7/24/2012	/BruceMargulies/
Date	Bruce Margulies
	Registration No. 64,175

BTM
Printed: July 24, 2012 (10:51am)
Y:\Clients\SCOT Scott A Moskowitz and Wistaria Trading, Inc\SCOT0014-6\Drafts\ApplicationFiling_SCOT0014-6_7-22-2012a.wpd

DATA PROTECTION METHOD AND DEVICE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a divisional of U.S. patent application Ser. No. 10/602,777, which is a continuation application of U.S. patent application Ser. No. 09/046,627 (which issued Jul. 22, 2003, as U.S. Pat. No. 6,598,162), which is a continuation-in-part of U.S. patent application Ser. No. 08/587,943, filed Jan. 17, 1996, (which issued Apr. 28, 1998, as U.S. Pat. No. 5,745,943). The entire disclosure of U.S. patent application Ser. No. 09/046,627 (which issued Jul. 22, 2003, as U.S. Pat. No. 6,598,162) and U.S. patent application Ser. No. 08/587,943, filed Jan. 17, 1996, (which issued Apr. 28, 1998, as U.S. Pat. No. 5,745,943) are hereby incorporated by reference in their entireties.

FIELD OF THE INVENTION

[0002] The invention relates to the protection of digital information. More particularly, the invention relates to a method and device for data protection.

[0003] With the advent of computer networks and digital multimedia, protection of intellectual property has become a prime concern for creators and publishers of digitized copies of copyrightable works, such as musical recordings, movies, video games, and computer software. One method of protecting copyrights in the digital domain is to use "digital watermarks."

[0004] The prior art includes copy protection systems attempted at many stages in the development of the software industry. These may be various methods by which a software engineer can write the software in a clever manner to determine if it has been copied, and if so to deactivate itself. Also included are undocumented changes to the storage format of the content. Copy protection was generally abandoned by the software industry, since pirates were generally just as clever as the software engineers and figured out ways to modify the software and deactivate the protection. The cost of developing such protection was not justified considering the level of piracy which occurred despite the copy protection.

[0005] Other methods for protection of computer software include the requirement of entering certain numbers or facts that may be included in a packaged software's manual, when prompted at start-up. These may be overcome if copies of the manual are distributed to unintended users, or by patching the code to bypass these measures. Other methods include requiring a user to contact the software vendor and to receive "keys" for unlocking software after registration attached to some payment scheme, such as credit card authorization. Further methods include network-based searches of a user's hard drive and comparisons between what is registered to that user and what is actually installed on the user's general computing device. Other proposals, by such parties as AT&T's Bell Laboratories, use "kerning" or actual distance in pixels, in the rendering of text documents, rather than a varied number of ASCII characters. However, this approach can often be defeated by graphics processing analogous to sound processing, which randomizes that information. All of these methods require outside determination and verification of the validity of the software license.

[0006] Digital watermarks can be used to mark each individual copy of a digitized work with information identifying the title, copyright holder, and even the licensed owner of a particular copy. When marked with licensing and ownership information, responsibility is created for individual copies where before there was none. Computer application programs can be watermarked by watermarking digital content resources used in conjunction with images or audio data Digital watermarks can be encoded with random or pseudo random keys, which act as secret maps for locating the watermarks. These keys make it impossible for a party to find the watermark without having the key. In addition, the encoding method can be enhanced to force a party to cause damage to a watermarked data stream when trying to erase a random-key watermark. Other information is disclosed in "Technology: Digital Commerce", Denise Caruso, New York Times, Aug. 7, 1995; and "Copyrighting in the Information Age", Harley Ungar, ONLINE MARKETPLACE, September 1995, Jupiter Communications.

[0007] Additionally, other methods for hiding information signals in content signals, are disclosed in U.S. Pat. No. 5,319,735--Preuss et al. and U.S. Pat. No. 5,379,345--Greenberg.

[0008] It is desirable to use a "stega-cipher" or watermarking process to hide the necessary parts or resources of the executable object code in the digitized sample resources. It is also desirable to further modify the underlying structure of an executable computer application such that it is more resistant to attempts at patching and analysis by memory capture. A computer application seeks to provide a user with certain utilities or tools, that is, users interact with a computer or similar device to accomplish various tasks and applications provide the relevant interface. Thus, a level of authentication can also be introduced into software, or "digital products," that include digital content, such as audio, video, pictures or multimedia, with digital watermarks. Security is maximized because erasing this code watermark without a key results in the destruction of one or more essential parts of the underlying application, rendering the "program" useless to the unintended user who lacks the appropriate key. Further, if the key is linked to a license code by means of a mathematical function, a mechanism for identifying the licensed owner of an application is created.

[0009] It is also desirable to randomly reorganize program memory structure intermittently during program run time, to prevent attempts at memory capture or object code analysis aimed at eliminating licensing or ownership information, or otherwise modifying, in an unintended manner, the functioning of the application.

[0010] In this way, attempts to capture memory to determine underlying functionality or provide a "patch" to facilitate unauthorized use of the "application," or computer program, without destroying the functionality and thus usefulness of a copyrightable computer program can be made difficult or impossible.

[0011] It is thus the goal of the present invention to provide a higher level of copyright security to object code on par with methods described in digital watermarking systems for digitized media content such as pictures, audio, video and multimedia content in its multifarious forms, as described in previous disclosures, "Steganographic Method and Device" Ser. No. 08/489,172, filed Jun. 7, 1995, now U.S. Pat. No. 5,613,004, and "Human Assisted Random Key Generation and Application for Digital Watermark System", Ser. No. 08/587,944, filed on Jan. 17, 1996, the disclosure of which is hereby incorporated by reference.

[0012] It is a further goal of the present invention to establish methods of copyright protection that can be combined with such schemes as software metering, network distribution of code and specialized protection of software that is designed to work over a network, such as that proposed by Sun Microsystems in their HotJava browser and Java programming language, and manipulation of application code in proposed distribution of documents that can be exchanged with resources or the look and feel of the document being preserved over a network. Such systems are currently being offered by companies including Adobe, with their Acrobat software.

This latter goal is accomplished primarily by means of the watermarking of font, or typeface, resources included in applications or documents, which determine how a bitmap representation of the document is ultimately drawn on a presentation device.

[0013] The present invention includes an application of the technology of "digital watermarks." As described in previous disclosures, "Steganographic Method and Device" and "Human Assisted Random Key Generation and Application for Digital Watermark System," watermarks are particularly suitable to the identification, metering, distributing and authenticating digitized content such as pictures, audio, video and derivatives thereof under the description of "multimedia content." Methods have been described for combining both cryptographic methods, and steganography, or hiding something in plain view. Discussions of these technologies can be found in Applied Cryptography by Bruce Schneier and The Code Breakers by David Kahn. For more information on prior art public-key cryptosystems see U.S. Pat. No. 4,200,770 Diffie-Hellman, U.S. Pat. No. 4,218,582 Hellman, U.S. Pat. No. 4,405,829 RSA, U.S. Pat. No. 4,424,414 Hellman Pohlig. Computer code, or machine language instructions, which are not digitized and have zero tolerance for error, must be protected by derivative or alternative methods, such as those disclosed in this invention, which focuses on watermarking with "keys" derived from license codes or other ownership identification information, and using the watermarks encoded with such keys to hide an essential subset of the application code resources.

BACKGROUND OF THE INVENTION

[0014] Increasingly, commercially valuable information is being created and stored in "digital" form. For example, music, photographs and video can all be stored and transmitted as a series of numbers, such as 1's and 0's. Digital techniques let the original information be recreated in a very accurate manner. Unfortunately, digital techniques also let the information be easily copied without the information owner's permission.

[0015] Because unauthorized copying is clearly a disincentive to the digital distribution of valuable information, it is important to establish responsibility for copies and derivative copies of such works. For example, if each authorized digital copy of a popular song is identified with a unique number, any unauthorized copy of the song would also contain the number. This would allow the owner of the information, such as a song publisher, to investigate who made the unauthorized copy. Unfortunately, it is possible that the unique number could be erased or altered if it is simply tacked on at the beginning or end of the digital information.

[0016] As will be described, known digital "watermark" techniques give creators and publishers of digitized multimedia content localized, secured identification and authentication of that content. In considering the various forms of multimedia content, such as "master," stereo, National Television Standards Committee (NTSC) video, audio tape or compact disc, tolerance of quality will vary with individuals and affect the underlying commercial and aesthetic value of the content. For example, if a digital version of a popular song sounds distorted, it will be less valuable to users. It is therefore desirable to embed copyright, ownership or purchaser information, or some combination of these and related data, into the content in a way that will damage the content if the watermark is removed without authorization.

[0017] To achieve these goals, digital watermark systems insert ownership information in a way that causes little or no noticeable effects, or "artifacts," in the underlying content signal. For example, if a digital watermark is inserted into a digital version of a song, it is important that a listener not be bothered by the slight changes introduced by the watermark. It is also important for the watermark technique to maximize the encoding level and "location sensitivity" in the signal to force damage to the content signal when removal is attempted. Digital watermarks address many of these concerns, and research in the field has provided extremely robust and secure implementations.

[0018] What has been overlooked in many applications described in the art, however, are systems which closely mimic distribution of content as it occurs in the real world. For instance, many watermarking systems require the original un-watermarked content signal to enable detection or decode operations. These include highly publicized efforts by NEC, Digimarc and others. Such techniques are problematic because, in the real world, original master copies reside in a rights holders vaults and are not readily available to the public.

[0019] With much activity overly focused on watermark survivability, the security of a digital watermark is suspect. Any simple linear operation for encoding information into a signal may be used to erase the embedded signal by inverting the process. This is not a difficult task, especially when detection software is a plug-in freely available to the public, such as with Digimarc. In general, these systems seek to embed cryptographic information, not cryptographically embed information into target media content.

[0020] Other methods embed ownership information that is plainly visible in the media signal, such as the method described in U.S. Pat. No. 5,530,739 to Braudaway et al. The system described in Braudaway protects a digitized image by encoding a visible watermark to deter piracy. Such an implementation creates an immediate weakness in securing the embedded information because the watermark is plainly visible. Thus, no search for the embedded signal is necessary and the watermark can be more easily removed or altered. For example, while certainly useful to some rights owners, simply placing the symbol "©" in the digital information would only provide limited protection. Removal by adjusting the brightness of the pixels forming the "©" would not be difficult with respect to the computational resources required.

[0021] Other relevant prior art includes U.S. Pat. Nos. 4,979,210 and 5,073,925 to Nagata et al., which encodes information by modulating an audio signal in the amplitude/time domain. The modulations introduced in the Nagata process carry a "copy/don't copy" message, which is easily found and circumvented by one skilled in the art. The granularity of encoding is fixed by the amplitude and frequency modulation limits required to maintain inaudibility. These limits are relatively low, making it impractical to encode more information using the Nagata process.

[0022] Although U.S. Pat. No. 5,661,018 to Leighton describes a means to prevent collusion attacks in digital watermarks, the disclosed method may not actually provide the security described. For-example, in cases where the watermarking technique is linear, the "insertion envelope" or "watermarking space" is well-defined and thus susceptible to attacks less sophisticated than collusion by unauthorized parties. Over-encoding at the watermarking encoding level is but one simple attack in such linear implementations. Another consideration not made by Leighton is that commercially-valuable content may already exist in a un-watermarked form somewhere, easily accessible to potential pirates, gutting the need for any type of collusive activity. Digitally signing the embedded signal with preprocessing of watermark data is more likely to prevent successful collusion. Furthermore, a "baseline" watermark as disclosed is quite subjective. It is simply described elsewhere in the art as the "perceptually significant" regions of a signal. Making a watermarking function less linear or inverting the insertion of watermarks would seem to provide the same benefit without the additional work required to create a "baseline" watermark. Indeed, watermarking algorithms should already be capable of defining a target insertion envelope or region without additional steps. What is evident is the Leighton patent does not allow for initial prevention of attacks on an embedded watermark as the content is visibly or audibly unchanged.

[0023] It is also important that any method for providing security also function with broadcasting media over networks such as the Internet, which is also referred to as "streaming." Commercial "plug-in" products such as RealAudio and RealVideo, as well as applications by vendors VDONet and Xtreme, are common in such network environments. Most digital watermark implementations focus on common file base signals and fail to anticipate the security of streamed signals. It is desirable that any protection scheme be able to function with a plug-in player without advanced knowledge of the encoded media stream.

[0024] Other technologies focus solely on file-based security. These technologies illustrate the varying applications for security that must be evaluated for different media and distribution environments. Use of cryptolopes or cryptographic containers, as proposed by IBM in its Cryptolope product, and InterTrust, as described in U.S. Pat. Nos. 4,827,508, 4,977,594, 5,050,213 and 5,410,598, may discourage certain forms of piracy. Cryptographic containers, however, require a user to subscribe to particular decryption software to decrypt data. IBM's InfoMarket and InterTrust's DigiBox, among other implementations, provide a generalized model and need proprietary architecture to function. Every user must have a subscription or registration with the party which encrypts the data. Again, as a form of general encryption, the data is scrambled or encrypted without regard to the media and its formatting. Finally, control over copyrights or other neighboring rights is left with the implementing party, in this case, IBM, InterTrust or a similar provider.

[0025] Methods similar to these "trusted systems" exist, and Cerberus Central Limited and Liquid Audio, among a number of companies, offer systems which may functionally be thought of as subsets of IBM and InterTrust's more generalized security offerings. Both Cerberus and Liquid Audio propose proprietary player software which is registered to the user and "locked" in a manner parallel to the locking of content that is distributed via a cryptographic container. The economic trade-off in this model is that users are required to use each respective companies' proprietary player to play or otherwise manipulate content that is downloaded. If, as is the case presently, most music or other media is not available via these proprietary players and more companies propose non-compatible player formats, the proliferation of players will continue. Cerberus and Liquid Audio also by way of extension of their architectures provide for "near-CD quality" but proprietary compression. This requirement stems from the necessity not to allow content that has near-identical data make-up to an existing consumer electronic standard, in Cerberus and Liquid Audio's case the so-called Red Book audio CD standard of 16 bit 44.1 kHz, so that comparisons with the proprietary file may not yield how the player is secured. Knowledge of the player's file format renders its security ineffective as a file may be replicated and played on any common player, not the intended proprietary player of the provider of previously secured and uniquely formatted content. This is the parallel weakness to public key crypto-systems which have gutted security if enough plain text and cipher text comparisons enable a pirate to determine the user's private key.

[0026] Many approaches to digital watermarking leave detection and decoding control with the implementing party of the digital watermark, not the creator of the work to be protected. A set of secure digital watermark implementations address this fundamental control issue forming the basis of key-based approaches. These are covered by the following patents and pending applications, the entire disclosures of which are hereby incorporated by reference: U.S. Pat. No. 5,613,004 entitled "Steganographic Method and Device" and its derivative U.S. patent application Ser. No. 08/775,216 (which issued Nov. 11, 1997, as U.S. Pat. No. 5,687,236), U.S. patent application Ser. No. 08/587,944 entitled "Human Assisted Random Key Generation and Application for Digital Watermark System" (which issued Oct. 13, 1998, as U.S. Pat. No. 5,822,432), U.S. patent application Ser. No. 08/587,943 entitled "Method for Stega-Cipher Protection of Computer Code" (which issued Apr. 28, 1998, as U.S. Pat. No. 5,748,569), U.S. patent application Ser. No. 08/677,435 entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data" (which issued Mar. 30, 1999, as U.S. Pat. No. 5,889,868) and U.S. patent application Ser. No. 08/772,222 entitled "Z-Transform Implementation of Digital Watermarks" (which issued Jun. 20, 2000, as U.S. Pat. No. 6,078,664). Public key crypto-systems are described in U.S. Pat. Nos. 4,200,770, 4,218,582, 4,405,829 and 4,424,414, the entire disclosures of which are also hereby incorporated by reference.

[0027] In particular, an improved protection scheme is described in "Method for Stega-Cipher

Protection of Computer Code," U.S. patent application Ser. No. 08/587,943 (which issued Apr. 28, 1998, as U.S. Pat. No. 5,748,569). This technique uses the key-based insertion of binary executable computer code within a content signal that is subsequently, and necessarily, used to play or otherwise manipulate the signal in which it is encoded. With this system, however, certain computational requirements, such as one digital player per digital copy of content, may be necessitated. For instance, a consumer may download many copies of watermarked content. With this technique, the user would also be downloading as many copies of the digital player program. While this form of security may be desirable for some applications, it is not appropriate in many circumstances.

[0028] Finally, even when digital information is distributed in encoded form, it may be desirable to allow unauthorized users to play the information with a digital player, perhaps with a reduced level of quality. For example, a popular song may be encoded and freely distributed in encoded form to the public. The public, perhaps using commonly available plug-in digital players, could play the encoded content and hear the music in some degraded form. The music may-sound choppy, or fuzzy or be degraded in some other way. This lets the public decide, based on the available lower quality version of the song, if they want to purchase a key from the publisher to decode, or "clean-up," the content. Similar approaches could be used to distribute blurry pictures or low quality video. Or even "degraded" text, in the sense that only authenticated portions of the text can be determined with the predetermined key or a validated digital signature for the intended message.

[0029] In view of the foregoing, it can be appreciated that a substantial need exists for a method allowing encoded content to be played, with degraded quality, by a plug-in digital player, and solving the other problems discussed above.

SUMMARY OF THE INVENTION

[0030] The disadvantages of the art are alleviated to a great extent by a method for combining transfer functions with predetermined key creation. In one embodiment, digital information, including a digital sample and format information, is protected by identifying and encoding a portion of the format information. Encoded digital information, including the digital sample and the encoded format information, is generated to protect the original digital information.

[0031] In another embodiment, a digital signal, including digital samples in a file format having an inherent granularity, is protected by creating a predetermined key. The predetermined key is comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

[0032] It is thus a goal of the present invention, to provide a level of security for executable code on similar grounds as that which can be provided for digitized samples. Furthermore, the present invention differs from the prior art in that it does not attempt to stop copying, but rather, determines responsibility for a copy by ensuring that licensing information must be preserved in descendant copies from an original. Without the correct license information, the copy cannot function.

[0033] An improvement over the art is disclosed in the present invention, in that the software itself is a set of commands, compiled by software engineer, which can be configured in such a manner as to tie underlying functionality to the license or authorization of the copy in possession by the user. Without such verification, the functions sought out by the user in the form of software cease to properly work. Attempts to tamper or "patch" substitute code resources can be made highly difficult by randomizing the location of said resources in memory on an intermittent basis to resist most attacks at disabling the system.

[0034] With these and other advantages and features of the invention that will become hereinafter apparent, the nature of the invention may be more clearly understood by reference to the following detailed description of the invention, the appended claims and to the several drawings attached herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0035] FIG. 1 is a block flow diagram of a method for copy protection or authentication of digital information according to an embodiment of the present invention.

DETAILED DESCRIPTION

[0036] In accordance with an embodiment of the present invention, a method combines transfer functions with predetermined key creation. Increased security is achieved in the method by combining elements of "public-key steganography" with cryptographic protocols, which keep in-transit data secure by scrambling the data with "keys" in a manner that is not apparent to those with access to the content to be distributed. Because different forms of randomness are combined to offer robust, distributed security, the present invention addresses an architectural "gray space" between two important areas of security: digital watermarks, a subset of the more general art of steganography, and cryptography. One form of randomness exists in the mask sets that are randomly created to map watermark data into an otherwise unrelated digital signal. The second form of randomness is the random permutations of data formats used with digital players to manipulate the content with the predetermined keys. These forms can be thought of as the transfer function versus the mapping function inherent to digital watermarking processes.

[0037] According to an embodiment of the present invention, a predetermined, or randomly generated, key is used to scramble digital information in a way that is unlike known "digital watermark" techniques and public key crypto-systems. As used herein, a key is also referred to as a "mask set" which includes one or more random or pseudo-random series of bits. Prior to encoding, a mask can be generated by any cryptographically secure random generation process. A block cipher, such as a Data Encryption Standard (DES) algorithm, in combination with a sufficiently random seed value, such as one created using a Message Digest 5 (MD5) algorithm, emulates a cryptographically secure random bit generator. The keys are saved in a database, along with information matching them to the digital signal, for use in descrambling and subsequent viewing or playback. Additional file format or transfer property information is prepared and made available to the encoder, in a bit addressable manner. As well, any authenticating function can be combined, such as Digital Signature Standard (DSS) or Secure Hash Algorithm (SHA).

[0038] Using the predetermined key comprised of a transfer function-based mask set, the data representing the original content is manipulated at the inherent granularity of the file format of the underlying digitized samples. Instead of providing, or otherwise distributing, watermarked content that is not noticeably altered, a partially "scrambled" copy of the content is distributed. The key is necessary both to register the sought-after content and to descramble the content into its original form.

[0039] The present invention uses methods disclosed in "Method for Stega-Cipher Protection of Computer Code," U.S. patent application Ser. No. 08/587,943 (which issued Apr. 28, 1998, as U.S. Pat. No. 5,748,569), with respect to transfer functions related to the common file formats, such as PICT, TIFF, AIFF, WAV, etc. Additionally, in cases where the content has not been altered beyond being encoded with such functional data, it is possible for a digital player to still play the content because the file format has not been altered. Thus, the encoded content could still be played by a plug-in digital player as discrete, digitally sampled signals, watermarked or not. That is, the structure of the file can remain basically unchanged by the watermarking process, letting common file format based players work with the "scrambled" content.

[0040] For example, the Compact Disc-Digital Audio (CD-DA) format stores audio information as a series of frames. Each frame contains a number of digital samples representing, for example, music, and a header that contains file format information. As shown in FIG. 1, according to an embodiment of the present invention some of the header information can be identified and "scrambled" using the predetermined key at steps 110 to 130. The music samples can remain unchanged. Using this technique, a traditional CD-DA player will be able to play a distorted version of the music in the sample. The amount of distortion will depend on the way, and extent, that the header, or file format, information has been scrambled. It would also be possible to instead scramble some of the digital samples while leaving the header information alone. In general, the digital signal would be protected by manipulating data at the inherent granularity, or "frames," of the CD-DA file format. To decode the information, a predetermined key is used before playing the digital information at steps 140 and 150.

[0041] A key-based decoder can act as a "plug-in" digital player of broadcast signal streams without foreknowledge of the encoded media stream. Moreover, the data format orientation is used to partially scramble data in transit to prevent unauthorized descrambled access by decoders that lack authorized keys. A distributed key can be used to unscramble the scrambled content because a decoder would understand how to process the key. Similar to on-the-fly decryption operations, the benefits inherent in this embodiment include the fact that the combination of watermarked content security, which is key-based, and the descrambling of the data, can be performed by the same key which can be a plurality of mask sets. The mask sets may include primary, convolution and message delimiter masks with file format data included.

[0042] The creation of an optimized "envelope" for insertion of watermarks provides the basis of much watermark security, but is also a complementary goal of the present invention. The predetermined or random key that is generated is not only an essential map to access the hidden information signal, but is also the descrambler of the previously scrambled signal's format for playback or viewing.

[0043] In a system requiring keys for watermarking content and validating the distribution of the content, different keys may be used to encode different information while secure one way hash functions or one-time pads may be incorporated to secure the embedded signal. The same keys can be used to later validate the embedded digital signature, or even fully decode the digital watermark if desired. Publishers can easily stipulate that content not only be digitally watermarked but that distributors must check the validity of the watermarks by performing digital signature-checks with keys that lack any other functionality. The system can extend to simple authentication of text in other embodiments.

[0044] Before such a market is economically feasible, there are other methods for deploying key-based watermarking coupled with transfer functions to partially scramble the content to be distributed without performing full public key encryption, i.e., a key pair is not necessarily generated, simply, a predetermined key's function is created to re-map the data of the content file in a lossless process. Moreover, the scrambling performed by the present invention may be more dependent on the file in question. Dissimilarly, encryption is not specific to any particular media but is performed on data. The file format remains unchanged, rendering the file useable by any conventional viewer/player, but the signal quality can be intentionally degraded in the absence of the proper player and key. Public-key encryption seeks to completely obscure the sensitive "plaintext" to prevent comparisons with the "ciphertext" to determine a user's private keys. Centralized encryption only differs in the utilization of a single key for both encryption and decryption making the key even more highly vulnerable to attacks to defeat the encryption process. With the present invention, a highly sought after photograph may be hazy to the viewer using any number of commonly available, nonproprietary software or hardware, without the authorized key. Similarly, a commercially valuable song may sound poor.

[0045] The benefit of some form of cryptography is not lost in the present invention. In fact, some piracy can be deterred when the target signal may be known but is clearly being protected through scrambling. What is not anticipated by known techniques, is an ala carte method to change various aspects of file formatting to enable various "scrambled states" for content to be subsequently distributed. An image may lack all red pixels or may not have any of the most significant bits activated. An audio sample can similarly be scrambled to render it less-than-commercially viable.

[0046] The present invention also provides improvements over known network-based methods, such as those used for the streaming of media data over the Internet. By manipulating file formats, the broadcast media, which has been altered to "fit" within electronic distribution parameters, such as bandwidth availability and error correction considerations, can be more effectively utilized to restrict the subsequent use of the content while in transit as well as real-time viewing or playing.

[0047] The mask set providing the transfer function can be read on a per-use basis by issuing an authorized or authenticating "key" for descrambling the signal that is apparent to a viewer or a player or possessor of the authenticating key. The mask set can be read on a per-computer basis by issuing the authorized key that is more generalized for the computer that receives the broadcast signals. Metering and subscription models become viable advantages over known digital watermark systems which assist in designating the ownership of a copy of digitized media content, but do not prevent or restrict the copying or manipulation of the sampled signal in question. For broadcast or streamed media, this is especially the case. Message authentication is also possible, though not guaranteeing the same security as an encrypted file as with general crypto systems.

[0048] The present invention thus benefits from the proprietary player model without relying on proprietary players. No new players will be necessary and existing multimedia file formats can be altered to exact a measure of security which is further increased when coupled with digital watermarks. As with most consumer markets for media content, predominant file formats exist, de facto, and corresponding formats for computers likewise exist. For a commercial compact disc quality audio recording, or 16 bit 44.1 kHz, corresponding file formats include: Audio Interchange File Format (AIFF), Microsoft WAV, Sound Designer II, Sun's .au, Apple's Quicktime, etc. For still image media, formats are similarly abundant: TIFF, PICT, JPEG, GIF, etc. Requiring the use of additional proprietary players, and their complementary file formats, for limited benefits in security is wasteful. Moreover, almost all computers today are multimedia-capable, and this is increasingly so with the popularity of Intel's MMX chip architecture and the PowerPC line of microchips. Because file formatting is fundamental in the playback of the underlying data, the predetermined key can act both as a map, for information to be encoded as watermark data regarding ownership, and a descrambler of the file that has been distributed. Limitations will only exist in how large the key must be retrofitted for a given application, but any manipulation of file format information is not likely to exceed the size of data required versus that for an entire proprietary player.

[0049] As with previous disclosures by the inventor on digital watermarking techniques, the present invention may be implemented with a variety of cryptographic protocols to increase both confidence and security in the underlying system. A predetermined key is described as a set of masks. These masks may include primary, convolution and message delimiter mask. In previous disclosures, the functionality of these masks is defined solely for mapping. The present invention includes a mask set which is also controlled by the distributing party of a copy of a given media signal. This mask set is a transfer function which is limited only by the parameters of the file format in question. To increase the uniqueness or security of each key used to scramble a given media file copy, a secure one way hash function can be used subsequent to transfer properties that are initiated to prevent the forging of a particular key. Public and private keys may be used as key pairs to further increase the unlikeliness that a key may be compromised.

[0050] These same cryptographic protocols can be combined with the embodiments of the present invention in administering streamed content that requires authorized keys to correctly display or play the streamed content in an unscrambled manner. As with digital watermarking, symmetric or asymmetric public key pairs may be used in a variety of implementations. Additionally, the need for certification authorities to maintain authentic key-pairs becomes a consideration for greater security beyond symmetric key implementations. The cryptographic protocols makes possible, as well, a message of text to be authenticated by a message authenticating function in a general computing device that is able to ensure secure message exchanges between authorizing parties.

[0051] An executable computer program is variously referred to as an application, from the point of view of a user, or executable object code from the point of view of the engineer. A collection of smaller, atomic (or indivisible) chunks of object code typically comprise the complete executable object code or application which may also require the presence of certain data resources. These indivisible portions of object code correspond with the programmers' function or procedure implementations in higher level languages, such as C or Pascal. In creating an application, a programmer writes "code" in a higher level language, which is then compiled down into "machine language," or, the executable object code, which can actually be run by a computer, general purpose or otherwise. Each function, or procedure, written in the programming language, represents a self-contained portion of the larger program, and implements, typically, a very small piece of its functionality. The order in which the programmer types the code for the various functions or procedures, and the distribution of and arrangement of these implementations in various files which hold them is unimportant. Within a function or procedure, however, the order of individual language constructs, which correspond to particular machine instructions is important, and so functions or procedures are considered indivisible for purposes of this discussion. That is, once a function or procedure is compiled, the order of the machine instructions which comprise the executable object code of the function is important and their order in the computer memory is of vital importance. Note that many "compilers" perform "optimizations" within functions or procedures, which determine, on a limited scale, if there is a better arrangement for executable instructions which is more efficient than that constructed by the programmer, but does not change the result of the function or procedure. Once these optimizations are performed, however, making random changes to the order of instructions is very likely to "break" the function. When a program is compiled, then, it consists of a collection of these sub-objects, whose exact order or arrangement in memory is not important, so long as any sub-object which uses another sub-object knows where in memory it can be found.

[0052] The memory address of the first instruction in one of these sub-objects is called the "entry point" of the function or procedure. The rest of the instructions comprising that sub-object immediately follow from the entry point. Some systems may prefix information to the entry point which describes calling and return conventions for the code which follows, an example is the Apple Macintosh Operating System (MacOS). These sub-objects can be packaged into what are referred to in certain systems as "code resources," which may be stored separately from the application, or shared with other applications, although not necessarily. Within an application there are also data objects, which consist of some data to be operated on by the executable code. These data objects are not executable. That is, they do not consist of executable instructions. The data objects can be referred to in certain systems as "resources."

[0053] When a user purchases or acquires a computer program, she seeks a computer program that "functions" in a desired manner. Simply, computer software is overwhelmingly purchased for its underlying functionality. In contrast, persons who copy multimedia content, such as pictures, audio and video, do so for the entertainment or commercial value of the content. The difference between the two types of products is that multimedia content is not generally interactive, but is instead passive, and its commercial value relates more on passive not interactive or utility features, such as those required in packaged software, set-top boxes, cellular phones, VCRs, PDAs, and the like. Interactive digital products which include computer code may be mostly interactive but can also contain content to add to the interactive experience of the user or make the underlying utility of the software more aesthetically pleasing. It is a common concern of both of these creators, both of interactive and passive multimedia products, that "digital products" can be easily and perfectly copied and made into unpaid or unauthorized copies. This concern is especially heightened when the underlying product is copyright protected and intended for commercial use.

[0054] The first method of the present invention described involves hiding necessary "parts" or code "resources" in digitized sample resources using a "digital watermarking" process, such as that described in the "Steganographic Method and Device" patent application. The basic premise for this scheme is that there are a certain sub-set of executable code resources, that comprise an application and that are "essential" to the proper function of the application. In general, any code resource can be considered "essential" in that if the program proceeds to a point where it must "call" the code resource and the code resource is not present in memory, or cannot be loaded, then the program fails. However, the present invention uses a definition of "essential" which is more narrow. This is because, those skilled in the art or those with programming experience, may create a derivative program, not unlike the utility provided by the original program, by writing additional or substituted code to work around unavailable resources. This is particularly true with programs that incorporate an optional "plug-in architecture," where several code resources may be made optionally available at run-time. The present invention is also concerned with concentrated efforts by technically skilled people who can analyze executable object code and "patch" it to ignore or bypass certain code resources. Thus, for the present embodiment's purposes, "essential" means that the function which distinguishes this application from any other application depends upon the presence and use of the code resource in question. The best candidates for this type of code resources are NOT optional, or plug-in types, unless special care is taken to prevent work-arounds.

[0055] Given that there are one or more of these essential resources, what is needed to realize the present invention is the presence of certain data resources of a type which are amenable to the "stega-cipher" process described in the "Steganographic Method and Device" patent U.S. Pat. No. 5,613,004. Data which consists of image or audio samples is particularly useful. Because this data consists of digital samples, digital watermarks can be introduced into the samples. What is further meant is that certain applications include image and audio samples which are important to the look and feel of the program or are essential to the processing of the application's functionality when used by the user. These computer programs are familiar to users of computers but also less obvious to users of other devices that run applications that are equivalent in some measure of functionality to general purpose computers including, but not limited to, set-top boxes, cellular phones, "smart televisions," PDAs and the like. However, programs still comprise the underlying "operating systems" of these devices and are becoming more complex with increases in functionality.

[0056] One method of the present invention is now discussed. When code and data resources are compiled and assembled into a precursor of an executable program the next step is to use a utility application for final assembly of the executable application. The programmer marks several essential code resources in a list displayed by the utility. The utility will choose one or several essential code resources, and encode them into one or several data resources using the stegacipher process. The end result will be that these essential code resources are not stored in their own partition, but rather stored as encoded information in data resources. They are not accessible at run-time without the key. Basically, the essential code resources that provide functionality in the final end-product, an executable application or computer program, are no longer easily and recognizably available for manipulation by those seeking to remove the underlying copyright or license, or its equivalent information, or those with skill to substitute alternative code resources to "force" the application program to run as an unauthorized copy. For the encoding of the essential code resources, a "key" is needed. Such a key is similar to those described in U.S. Pat. No. 5,613,004, the "Steganographic Method and Device" patent. The purpose of this scheme is to make a particular licensed copy of an application distinguishable from any other. It is not necessary to distinguish every instance of an application, merely every instance of a license. A licensed user may then wish to install multiple copies of an application, legally or with authorization. This method, then, is to choose the key so that it corresponds, is equal to, or is a function of, a license code or license descriptive information, not just a text file, audio clip or identifying piece of information as desired in digital watermarking schemes extant and typically useful to stand-alone, digitally sampled content. The key is necessary to access the underlying code, i.e., what the user understands to be the application program.

[0057] The assembly utility can be supplied with a key generated from a license code generated for the license in question. Alternatively, the key, possibly random, can be stored as a data resource and encrypted with a derivative of the license code. Given the key, it encodes one or several essential resources into one or several data resources. Exactly which code resources are encoded into which data resources may be determined in a random or pseudo random manner. Note further that the application contains a code resource which performs the function of decoding an encoded code resource from a data resource. The application must also contain a data resource which specifies in which data resource a particular code resource is encoded. This data resource is created and added at assembly time by the assembly utility. The application can then operate as follows:

[0058] 1) when it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. This can include a particular computer configuration;

[0059] 2) it stores this information in a personalization data resource;

[0060] 3) Once it has the license code, it can then generate the proper decoding key to access the essential code resources.

[0061] Note that the application can be copied in an uninhibited manner, but must contain the license code issued to the licensed owner, to access its essential code resources. The goal of the invention, copyright protection of computer code and establishment of responsibility for copies, is thus accomplished.

[0062] This invention represents a significant improvement over prior art because of the inherent difference in use of purely informational watermarks versus watermarks which contain executable object code. If the executable object code in a watermark is essential to an application which accesses the data which contains the watermark, this creates an all-or-none situation. Either the user must have the extracted watermark, or the application cannot be used, and hence the user cannot gain full access to the presentation of the information in the watermark bearing data. In order to extract a digital watermark, the user must have a key. The key, in turn, is a function of the license information for the copy of the software in question. The key is fixed prior to final assembly of the application files, and so cannot be changed at the option of the user. That, in turn, means the license information in the software copy must remain fixed, so that the correct key is available to the software. The key and the license information are, in fact, interchangeable. One is merely more readable than the other. In U.S. Pat. No. 5,613,004, the "Steganographic Method and Device, patent", the possibility of randomization erasure attacks on digital watermarks was discussed. Simply, it is always possible to erase a digital watermark, depending on how much damage you are willing to do to the watermark-bearing content stream. The present invention has the significant advantage that you must have the watermark to be able to use the code it contains. If you erase the watermark you have lost a key piece of the functionality of the application, or even the means to access the data which bear the watermark.

[0063] A preferred embodiment would be implemented in an embedded system, with a minimal operating system and memory. No media playing "applets," or smaller sized applications as proposed in new operating environments envisioned by Sun Microsystems and the advent of Sun's Java operating system, would be permanently stored in the system, only the bare necessities to operate the device, download information, decode watermarks and execute the applets contained in them. When an applet is finished executing, it is erased from memory. Such a system would guarantee that content which did not contain readable watermarks could not be used. This is a powerful control mechanism for ensuring that content to be distributed through such a system contains valid watermarks. Thus, in such networks as the Internet or set-top box controlled cable systems, distribution and exchange of content would be made more secure from unauthorized copying to the benefit of copyright holders and other related parties. The system would be enabled to invalidate, by default, any content which has had its watermark(s) erased, since the watermark conveys, in addition to copyright information, the means to fully access, play, record or otherwise manipulate, the content.

[0064] A second method according to the present invention is to randomly re-organize program memory structure to prevent attempts at memory capture or object code analysis. The object of this method is to make it extremely difficult to perform memory capture-based analysis of an executable computer program. This analysis is the basis for a method of attack to defeat the system envisioned by the present invention.

[0065] Once the code resources of a program are loaded into memory, they typically remain in a fixed position, unless the computer operating system finds it necessary to rearrange certain portions of memory during "system time," when the operating system code, not application code, is running. Typically, this is done in low memory systems, to maintain optimal memory utilization. The MacOS for example, uses Handles, which are double-indirect pointers to memory locations, in order to allow the operating system to rearrange memory transparently, underneath a running program. If a computer program contains countermeasures against unlicensed copying, a skilled technician can often take a snapshot of the code in memory, analyze it, determine which instructions comprise the countermeasures, and disable them in the stored application file, by means of a "patch." Other applications for designing code that moves to prevent scanning-tunnelling microscopes, and similar high sensitive hardware for analysis of electronic structure of microchips running code, have been proposed by such parties as Wave Systems. Designs of Wave Systems' microchip are intended for preventing attempts by hackers to "photograph" or otherwise determine "burn in" to microchips for attempts at reverse engineering. The present invention seeks to prevent attempts at understanding the code and its organization for the purpose of patching it. Unlike systems such as Wave Systems', the present invention seeks to move code around in such a manner as to complicate attempts by software engineers to reengineer a means to disable the methods for creating licensed copies on any device that lacks "trusted hardware." Moreover, the present invention concerns itself with any application software that may be used in general computing devices, not chipsets that are used in addition to an underlying computer to perform encryption. Wave Systems' approach to security of software, if interpreted similarly to the present invention, would dictate separate microchip sets for each piece of application software that would be tamperproof. This is not consistent with the economics of software and its distribution.

[0066] Under the present invention, the application contains a special code resource which knows about all the other code resources in memory. During execution time, this special code resource, called a "memory scheduler," can be called periodically, or at random or pseudo random intervals, at which time it intentionally shuffles the other code resources randomly in memory, so that someone trying to analyze snapshots of memory at various intervals cannot be sure if they are looking at the same code or organization from one "break" to the next. This adds significant complexity to their job. The scheduler also randomly relocates itself when it is finished. In order to do this, the scheduler would have to first copy itself to a new location, and then specifically modify the program counter and stack frame, so that it could then jump into the new copy of the scheduler, but return to the correct calling frame. Finally, the scheduler would need to maintain a list of all memory addresses which contain the address of the scheduler, and change them to reflect its new location.

[0067] The methods described above accomplish the purposes of the invention--to make it hard to analyze captured memory containing application executable code in order to create an identifiable computer program or application that is different from other copies and is less susceptible to unauthorized use by those attempting to disable the underlying copyright protection system. Simply, each copy has particular identifying information making that copy different from all other copies.

[0068] Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.

WHAT IS CLAIMED IS:

1. A method for copy protection of digital information, the digital information including a digital sample and format information, comprising the steps of: identifying a portion of the format information to be encoded; generating encoded format information from the identified portion of the format information; and generating encoded digital information, including the digital sample and the encoded format information.

2. The method of claim 1, further comprising the step of requiring a predetermined key to decode the encoded format information.

3. The method of claim 2, wherein the digital sample and format information are configured to be used with a digital player, and wherein information output from the digital player will have a degraded quality unless the encoded format information is decoded with the predetermined key.

4. The method of claim 3, wherein the information output from the digital player represents a still image, audio or video.

5. The method of claim 3, wherein the information output represents text data to be authenticated.

6. A method for protecting a digital signal, the digital signal including digital samples in a file format having an inherent granularity, comprising the step of:

creating a predetermined key comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

7. The method of claim 6, wherein the digital signal represents a continuous analog waveform.

8. The method of claim 6, wherein the predetermined key comprises a plurality of mask sets.

9. The method of claim 6, wherein the digital signal is a message to be authenticated.

10. The method of claim 6, wherein the mask set is ciphered by a key pair comprising a public key and a private key.

11. The method of claim 6, further comprising the step of:

using a digital watermarking technique to encode information that identifies ownership, use, or other information about the digital signal, into the digital signal.

12. The method of claim 6, wherein the digital signal represents a still image, audio or video.

13. The method of claim 6, further comprising the steps of:

selecting the mask set, including one or more masks having random or pseudo-random series of bits; and

validating the mask set at the start of the transfer function-based mask set.

14. The method of claim 13, wherein said step of validating comprises the step of: comparing a hash value computed at the start of the transfer function-based mask set with a determined transfer function of the hash value.

15. The method of claim 6, further comprising the steps of:

selecting the mask set, including one or more masks having random or pseudo-random series of bits; and

authenticating the mask set by comparing a hash value computed at the start of the transfer function-based mask set with a determined transfer function of the hash value.

16. The method of claim 13, wherein said step of validating comprises the step of: comparing a digital signature at the start of the transfer function-based mask set with a determined transfer function of the digital signature.

17. The method of claim 6, further comprising the steps of:

selecting the mask set, including one or more masks having random or pseudo-random series of bits; and

authenticating the mask set by comparing a digital signature at the start of the transfer function-based mask set with a determined transfer function of the digital signature.

18. The method of claim 13, further comprising the step of:

using a digital watermarking technique to embed information that identifies ownership, use, or other infonnation about the digital signal, into the digital signal; and

wherein said step of validating is dependent on validation of the embedded information.

19. The method of claim 6, further comprising the step of:

computing a secure one way hash function of carrier signal data in the digital signal, wherein the hash function is insensitive to changes introduced into the carrier signal for the purpose of carrying the transfer function-based mask set.

20. A method for protecting a digital signal, the digital signal including digital samples in a file format having an inherent granularity, comprising the steps of:

creating a predetermined key comprised of a transfer function-based mask set that can manipulate data at the inherent granularity of the file format of the underlying digitized samples;

authenticating the predetermined key containing the correct transfer function-based mask set during playback of the data; and

metering the playback of the data to monitor content.

21. The method of claim 20, wherein the predetermined key is authenticated to authenticate message information.

22. A method to prepare for the scrambling of a sample stream of data, comprising the steps of:

generating a plurality of mask sets to be used for encoding, including a random primary mask, a random convolution mask and a random start of message delimiter;

obtaining a transfer function to be implemented;

generating a message bit stream to be encoded;

loading the message bit stream; a stega-cipher map truth table, the primary mask, the convolution mask and the start of message delimiter into memory;

initializing the state of a primary mask index, a convolution mask index, and a message bit index; and

setting a message size equal to the total number of bits in the message bit stream.

23. A method to prepare for the encoding of stega-cipher information into a sample stream of data, comprising the steps of:

generating a mask set to be used for encoding, the set including a random primary mask, a random convolution mask, and a random start of message -delimiter;

obtaining a message to be encoded;

compressing and encrypting the message if desired;

generating a message bit stream to be encoded;

loading the message bit stream, a stega-cipher map truth table, the primary mask, the convolution mask and the start of message delimiter into memory;

initializing the state of a primary mask index, a convolution mask index, and a message bit index; and

setting the message size equal to the total number of bits in the message bit stream.

24. The method of claim 23 wherein the sample stream of data has a plurality of windows, further comprising the steps of:

calculating over which windows in the sample stream the message will be encoded;

computing a secure one way hash function of the information in the calculated windows, the hash function generating hash values insensitive to changes in the samples induced by a stega-cipher; and

encoding the computed hash values in an encoded stream of data.

25. The method of claim 13, wherein said step of selecting comprises the steps of:

collecting a series of random bits derived from keyboard latency intervals in random typing;

processing the initial series of random bits through an MD5 algorithm;

using the results of the MD5 processing to seed a triple-DES encryption loop;

cycling through the triple-DES encryption loop, extracting the least significant bit of each result after each cycle; and

concatenating the triple-DES output bits into the random series of bits.

26. A method for copy protection of digital information, the digital information including a digital sample and format information, comprising the steps of:

identifying a portion of the digital sample to be encoded;

generating an encoded digital sample from the identified portion of the digital sample; and

generating encoded digital information, including the encoded digital sample and the format information.

27. The method of claim 26, further comprising the step of requiring a predetermined key to decode the encoded digital sample.

28. The method of claim 27, wherein the digital sample and format information are configured to be used with a digital player. and wherein information output from the digital player will have a degraded quality unless the encoded digital sample is decoded with the predetermined key.

29. The method of claim 27, wherein information output will have non authentic message data unless the encode digital sample is decoded with the predetermined key.

30. A method for protecting digital data, where the digital data signal is organized into a plurality of frames, each frame having i) a header comprising file format information and ii) at least a portion of the digital data, said method comprising the steps of:

creating a predetermined key to manipulate the file format information in one or more of the plurality of frames; and

manipulating the file format information using the predetermined key in at least two of the plurality of frames, such that the digital data will be perceived by a human as noticeably altered if it is played without using a decode key to restore the file format information to a prior state.

31. The method of claim 30, wherein the predetermined key comprises a private key that is associated with a key pair.

32. A method for copy protection of software comprising: embedding the software with a

watermark wherein the embedded software operates in a manner substantially the same as the software prior to the embedding step.

33. The process of claim 32, wherein the step of embedding the software with a watermark increases the complexity of code analysis and/or tampering with the software.

34. The process of claim 32, wherein the watermarked software queries a user for personalization information during installation of the software.

35. The process of claim 32, wherein the watermark is accessible with a key.

36. The process of claim 35, wherein the key enables authorized use of the watermarked software.

37. The process according to claim 35, wherein the key and license information are interchangeable.

38. The process according to claim 32, wherein the step of embedding the software with a watermark is performed during execution of the software.

39. The process according to claim 32, wherein the step of embedding the software with a watermark modifies the structure of the software being embedded.

40. An article of manufacture comprising a machine readable medium, having thereon stored instructions adapted to be executed by a processor, which instructions when executed result in a process comprising: receiving potentially watermarked software; and identifying the software by extracting the watermark.

41. The article of manufacture of claim 40, wherein the watermark is associated with information fixed prior to distribution of the watermarked software.

42. The article of manufacture of claim 40, wherein the watermark affects functionality of the watermarked software.

43. The article of manufacture of claim 40, wherein the extracted watermark enables generation of a key.

44. The article of manufacture of claim 43, wherein the generated key and licensing information are associated.

45. The article of manufacture of claim 40, further comprising limiting functionality of the software if the watermark cannot be extracted.

46. A method for watermarking software comprising: determining the structure a plurality of code contained in the software; and configuring at least a portion of the plurality of code according to a watermarking process.

47. The process of claim 46, wherein the watermarking process further comprises inserting information into the software after installation.

48. The process of claim 46, wherein the watermarking process configures the at least a portion of the plurality of code according to a key.

49. The process of claim 46, wherein the watermarking process increases the complexity of code analysis and/or tampering with the software.

50. The process of claim 46, wherein the watermarking process is selected from the group comprising: data hiding, steganography or steganographic ciphering.

51. The process of claim 46, wherein the watermarking process is applied during execution of the software.

52. A system for copy protection of software comprising the steps of: associating license information with a copy of a software application; encoding the associated license information into the copy of the software application using a watermarking process; providing the copy of the software application having license information encoded therein to a user; and, comparing information received by a user with the encoded license information.

53. The system of claim 52, wherein the encoding is controlled by a key.

54. The system of claim 52, wherein the step of comparing the user supplied information with the encoded license information enables authorization of the software.

55. The system of claim 53, wherein the key is fixed prior to distribution of the software.

56. The system of claim 52, wherein the license information comprises code which affects functionality of the watermarked software.

57. The system of claim 52, wherein the watermarked software is resistant to code analysis and/or tampering.

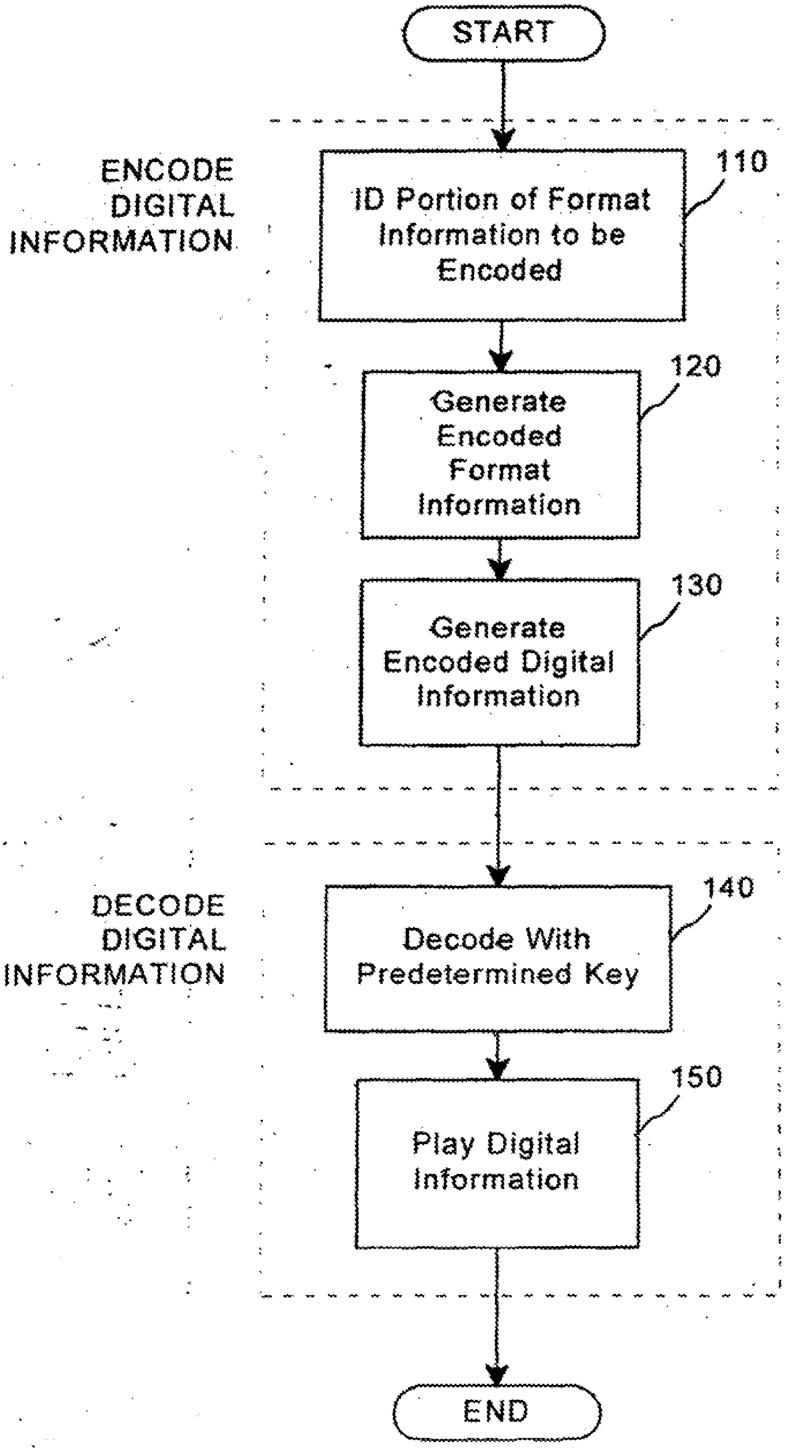

START
ENCODE DIGITAL INFORMATION
ID Portion of Format Information to be Encoded
110
Generate Encoded Format Information
120
Generate Encoded Digital Information
130
DECODE DIGITAL INFORMATION
Decode With Predetermined Key
140
Play Digital Information
150
END

FIG. 1

Attachment 16 Page 37 of 45

DECLARATION FOR PATENT APPLICATION

As one of the below named inventors, I hereby declare that:

My residence, post office address and citizenship is as stated below next to my name;

I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

DATA PROTECTION METHOD AND DEVICE

the specification of which: ☒ is attached hereto.
□ was filed on:
as Application No.:
and was amended on: ____________.

I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to patentability as defined in 37 C.F.R. § 1.56.

Prior Foreign Application(s)

I hereby claim foreign priority benefits under Title 35, United States Code, § 119(a)-(d) or § 365(b) of any foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed	
				Yes □	No □
				Yes □	No □

Prior Provisional Application(s)

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below:

Application Number	Date of Filing (day, month, year)

Prior United States Application(s)

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s), or § 365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Number	Date of Filing (day, month, year)	Status – Patented, Pending, Abandoned
10/602,777	June 25, 2003	Pending
09/046,627	March 24, 1998	Patent No. 6,598,162 July 22, 2003
08/587,943	January 17, 1996	Patent No. 5,745,569 April 28, 1998

All correspondence and telephone communications should be addressed to:

SCOTT MOSKOWITZ
16711 COLLINS AVENUE
NO. 2505
SUNNY ISLES BEACH, FLORIDA 33160

TELEPHONE NUMBER: (305) 956 - 9041
FACSIMILE NUMBER: (305) 956 - 9042

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine and imprisonment, or both, under 18 U.S.C. § 1001, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature ______________________ Date AUGUST 1, 2007

Full Name of First Inventor:	MOSKOWITZ (Family Name)	Scott (First Given Name)	A. (Second Given Name)

Citizenship: United States of America

Residence: 16711 Collins Avenue, # 2505, Sunny Isles Beach, FL 33160

Post Office Address: 16711 Collins Avenue, # 2505, Sunny Isles Beach, FL 33160

ABSTRACT OF THE DISCLOSURE

An apparatus and method for encoding and decoding additional information into a digital information in an integral manner. More particularly, the invention relates to a method and device for data protection.

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	Data protection method and device
First Named Inventor/Applicant Name:	Scott Moskowitz
Filer:	Bruce Talbot Margulies
Attorney Docket Number:	SCOT0014-6

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Utility application filing	1011	1	380	380
Utility Search Fee	1111	1	620	620
Utility Examination Fee	1311	1	250	250
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**1250**

Electronic Acknowledgement Receipt

EFS ID:	13323484
Application Number:	13556420
International Application Number:	
Confirmation Number:	5811
Title of Invention:	Data protection method and device
First Named Inventor/Applicant Name:	Scott Moskowitz
Customer Number:	31518
Filer:	Bruce Talbot Margulies
Filer Authorized By:	
Attorney Docket Number:	SCOT0014-6
Receipt Date:	24-JUL-2012
Filing Date:	
Time Stamp:	13:02:05
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$1250
RAM confirmation Number	12992
Deposit Account	
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)

1		ApplicationFiling_SCOT0014-6_7-22-2012c.pdf	563323 [illegible]	yes	41

Multipart Description/PDF files in .zip description

Document Description	Start	End
Transmittal of New Application	1	2
Preliminary Amendment	3	6
Specification	7	28
Claims	29	36
Abstract	37	37
Drawings-only black and white line drawings	38	38
Oath or Declaration filed	39	41

Warnings:

Information:

2	Fee Worksheet (SB06)	fee-info.pdf	32740 [illegible]	no	2

Warnings:

Information:

Total Files Size (in bytes):	596063

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

ATTACHMENT 17

This is attachment 17. This attachment is part of the response to the office action.

The section below for Chart 1 shows support for the Claims 1, 2, 3, 4, 5, 8, l0, and 12 of USP 9021602 in prior application 08/587,943.

The sections below for Charts 2-5 show the same support, that is the same descriptive text as in application 08/587,943, either by incorporation or expressly present, in the four benefit applications claimed in USP 9021602.

Moreover, each of the four applications also issued as a patent as shown by the cover page of USP 9021602 and the cover pages of each of these four patents, which are Attachments 19-22.

Finally, the section below for chart 6 shows the same descriptive support appearing in 08/587,943, for Claims 1, 2, 3, 4, 5, 8, l0, and 12 of USP 9021602 is also present in USP 9021602.

CHART 1: Claims 1, 2, 3, 4, 5, 8, l0, and 12 of USP 9021602 and support in application 08/587,943 application, Attachment 1.

Chart 1 generally copies text of the corresponding chart in Mr. Moskowitz's 131 declaration.

Column 1 recites the claims and column 2 point cites the location of support in 08/587,943 and quotes relevant text from 08/587,943 and explains relevance.

CHART 1	
Rejected Claim of **USP 9021602**	Support in my 08/587,943 application, **Attachment 1.**
1. A computer based method for accessing functionality provided by an application software comprising:	Page 18:6-8 "The key is necessary to access the underlying code, i.e., what the user understands to be the application program." This statement follows my explanation at page 17:7 to 18:6 regarding how the software is compiled to encode certain code resources deemed "essential" for the functionality of the software in data resources. Page 11:24 to page 12:2 then explains how a user used the software including the functionality. That is how to perform the method of the preamble of this claim 1.

storing said application software in non transient memory of a computer;	On page 8:37 I refer to the program being "loaded" which means copied from slow memory (like disc) to fast memory (like RAM). Both of those forms (slow and fast) memory are "non transient." I understand that "non transient" were words the USPTO recommended everyone use in reference to memory when some court decision stated that memory might read on a "signal" and might be considered not patentable subject matter. I am told that the USPTO therefore took a "liberal" view of support for "non transient", basically allowing anyone claiming something stored in memory in a patent application that was not disclosing signals as memory, to add "non transient" to avoid adverse court invalidity determinations.
said application software in said computer prompting a user to enter into said computer personalization information;	On page 1:25-28, I point out that it was well known for computer software to prompt a use for information at startup. At page 11:25-28, I disclose that my software prompts the user to enter personalization information when run for the first time. Page 1:25-28 states "The application can then operate as follows: 1) when it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. This can include a particular computer configuration;."
said application software storing, in said non transient memory, in a personalization data resource, both computer configuration information of said computer, and a license code	On page 1:25-28, I continue on the next two lines (29, 30), to state "2) *it* stores *this* information in a personalization data resource;" The "it" is the software. The "this information" refers to the personalization information which lines 27 and 28 state may include both license code and computer configuration.
entered in response to said prompting;	That statement that item "2)" follows immediately after item "1)" in the sentence explaining operation of the software indicates that the storing of this information is in response to the entering of the information by the user.
said application software in said computer generating a proper decoding key,	At page 11, lines 31-33, I state "3) Once it has the license code, it can then generate the proper decoding key to access the essential code resources." This shows generating the proper decoding key.

said generating comprising using said license code; and	In the very next paragraph, at page 11:34-37, I state "Note that the application can be copied in an uninhibited manner, but must contain the license code issued to the licensed owner, to access its essential code resources." This indicates that the license code is essential to access the code resource, and accessing requires the decoding key, which indicates that the license code is also essential to generating the decoding key. However, I explained the significance of the key and that generating the key requires the license code, back in page 10, when describing how the software is assembled. "This method, then, is to choose the key so that it corresponds, is equal to, or is a function of, a license code or license descriptive information." In other words, I disclosed that generating the decode key may requires the licence code.
wherein said application software, in said computer, cannot access at least one encoded code resource of said application software, unless said license code is stored in said personalization data resource.	At page 10:16-20, I states that "The end result [of compiling the software is that]...these essential code resources... are not accessible at run-time, without the key." This the same thing as stated by this wherein clause in claim 1.
2. The method of claim 1, wherein said encoded code resource is encoded in at least one data resource.	In describing the assembly utility, at page 11:9-15, I state that the assembly utility "encodes one or several essential resources into one or several data resources." At page 10:14-16, I also state that "The utility will chose one or several essential code resources, and encode them into one or several data resources." This discloses that "code resource" are "encoded" and that they are encoded in at least one data resource, as claimed.

3. The method of claim 1 wherein said encoded code resource is steganographically encoded.	At page 5:9-10, I explain that "steganography" refers to "hiding something in plain view." At page 8:25-27, I disclose that the first method of my invention "involves hiding necessary 'parts' or code 'resources' in digitized sample resources. At page 3:10-13, I explain that "It is desirable to use a 'stega-cipher' .... process to hide necessary parts of resource of executable object code in the digitized sample resource." At page 10:15-16, I discloses that the encoding may be "using the stegacipher process." I think these support use of the adverb "steganographically" when referring to encoding that uses steganography for hiding the code resources in data resources.
4. The method of claim 3 wherein said encoded code resource is encoded in a data resource.	Claim 4 is the same as claim 2, but depends upon claim 3 instead of claim 1. So my discussion of claim 2 applies here.
5. The method of claim 1 wherein said computer configuration information is stored in a data resource.	In my description of the operation of the software application at page 11:27-30, I stated "This can include a particular computer configuration; 2) it stores *this information* in a personalization *data resource*." That is, I clearly stated that the computer configuration information may be stored in a data resource.

8. The method of claim 1 wherein said computer comprises a processor and said application software using said processor in said prompting and said storing.	My disclosure in Attachment 1 is replete with references to computers in the context of digital computer which necessarily convey the presence of a processor. For example, in discussing the structure of software, I refer to "the instructions" (page 7:21) which immediately conveys a processor for acting on instruction. Similarly, I refer to "executable code" (page 7:32) which immediately conveys a processor for acting on the code. At page 11, in describing a software application including essential code resources encoded in data resource, I state "The application can then operate as follows: 1) when it is run for the first time...." Running is a colloquial expression for a digital computer executing instructions in a software program. Digital computers necessarily include a processor. Similarly, at pages 12-13, I describe a preferred embodiment as implemented in an embedded system with a minimal operating system. Further, at page 14:25-27, I stated that "the present invention concerns itself with any application software that may be used in general computing devices." The term "general computer devices" immediately conveys a processor for use by application software for both prompting (an I/O function) and storing (a data storage function). Finally, the original claims defined the step of "processing" of data, which discloses a processor. See page 17:4 (claim 4).
10. A computer program product storing in a non transitory storage media computer application software code for an application software product, which, when run by a computer system, causes said computer system to perform the following for accessing functionality provided by said application software product, comprising:	This is a description of software stored on some physical medium. In the Background section in Attachment 1, I describe that software may be stored on a user's hard drive, when referring to attempts to enforce licencing. That is, at page 2, I state "Further methods include network-based searches of a user's hard drive and comparisons between what is registered to that user and what is actually installed on the user's general computing device." At page 10:36, I refer to "install[ed] ... copies," which refers to installation on a drive. A computer's drive is a product. At page 3:32-33, I refer to storing code in computer memory, stating "It is also desirable to randomly reorganize program memory structure intermittently during program run time." At page 7:2-5, I refer to "the order of the machine instructions. .. In the computer memory." Computer memory is a product.

storing said application software code in non transient memory of a computer system; said application software code in said computer system prompting a user to enter into said computer system personalization information; said application software code storing, in said non transient memory, in a personalization data resource, both computer configuration information of said computer system, and a license code entered in response to said prompting; said application software code in said computer system generating a proper decoding key, said generating comprising using said license code; and wherein said application software code, in said computer system, cannot access at least one encoded code resource of said application software code, unless said license code is stored in said personalization data resource.	This is the same recitation appearing in claim 1. See my discussion of claim 1 herein above.
12. The product of claim 10 wherein said computer program product causes storing of said encoded code resource in a data resource in non transient memory of said computer.	Claim 12 contains the same recitation ("storing of said encoded code resource in a data resource ") as claim 2. Attachment 1 shows I disclosed this feature for the reasons stated for claim 1.

CHART 2: Claims 1, 2, 3, 4, 5, 8, l0, and 12 of USP 9021602 and the same support in application 09/046,627 Attachment 13, as quoted above for application 08/587,943, Attachment 1

Chart 2 shows support for the claims based upon incorporation by reference of the disclosure of 08/587,943. As noted in Chart 1, application 08/587,943 supports these claims. Therefore, the incorporation by reference of the disclosure of 08/587,943 into application 09/046,627 shows that application 09/046,627 also supports these claims.

Moreover, application 09/046,627 attachment 13, at page 1:12-14 also expressly claimed benefit to application 08/587,943.

CHART 2	
Rejected Claim of USP 9021602	Support in application **09/046,627 Attachment 13**
Claims 1, 2, 3, 4, 5, 8, l0, and 12 of USP 9021602	Page 1:12-14 “This application claims the benefit of U.S. patent application Serial No. 08/587,943, filed January 17, 1996, entitled "Method for Stega-Cipher Protection of Computer Code," the entire disclosure of which is hereby incorporated by reference.” Page 7:6-14 “These are covered by the following patents and pending applications, the entire disclosures of which are **hereby incorporated by reference**: US Patent No. 5,613,004 entitled "Steganographic Method and Device" and its derivative US patent application Serial No. 081775,216, US patent application Serial No. 08/587,944 entitled "Human Assisted 10 . Random Key Generation and Application for DigitaI Watermark System," US Patent Application Serial No. **08/587,943** entitled "Method for Stega-Cipher Protection of Computer Code," US patent application Serial No. 08/677,435 entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data," and US Patent Application Serial No. 081772,222 entitled "2-Transform Implementation of Digital Watermarks."

CHART 3: Claims 1, 2, 3, 4, 5, 8, l0, and 12 of USP 9021602 and the same support in application 10/602,777 Attachment 14

Chart 3 shows support for the rejected claims of USP 9021602 based upon incorporation by reference of the disclosure of 08/587,943. As noted in Chart 1, application 08/587,943 supports these claims. Therefore, the incorporation by reference of the disclosure of 08/587,943 into application 10/602,777 shows that application 10/602,777 also supports these claims.

Moreover, application 10/602,777 originally claimed benefit to application 08/587,943, at Attachment 14, pdf page 20, lines 11-4. However, this claim to benefit was deleted by the preliminary amendment.

The preliminary amendment, also filed 2003-06-25 in application 10/602,777, at attachment 14, pdf page 11:2-9 stated:

> Please delete the section entitled "CROSS-REFERENCE TO RELATED APPLICATIONS" on page 1, lines 10-14, of the originally filed application and insert the new section entitled CROSS-REFERENCE TO RELATED APPLICATIONS" on page 1, at line 5:
>
> --This application is a continuation application of U.S. Patent Application Serial No. 09/046,627 (now awaiting issuance), which is a continuation of U.S. Patent " Application Serial No. **08/587,943**, filed January 17, 1996, (which issued April 28, 1998, as U.S. Patent No. 5,745,943). The entire disclosure of U.S. Patent Application No. 09/046,627 is hereby incorporated by reference.--

Attachment 14, at pdf page 26 contains part of the specification, as filed. This pdf page 26:6-16 states:

> These are covered by the following patents and pending applications, the entire disclosures of which **are hereby incorporated by reference**: US Patent No. 5,613,004 entitled "Steganographic Method and Device" and its derivative US patent application Serial No. 081775,216, US patent application Serial No. 08/587,944 entitled "Human Assisted 10 . Random Key Generation and Application for DigitaI Watermark System," US Patent Application Serial No. **08/587,943** entitled "Method for Stega-Cipher Protection of Computer Code," US patent application Serial No. 08/677,435 entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data," and US Patent Application Serial No. 08/772,222 entitled "2-Transform Implementation of Digital Watermarks." Public key crypto-systems are described in US Patents No. 4,200,770, 4,218,582, 4,405,829 and 4,424,414, the entire disclosures of which are also hereby incorporated by reference.

This contains an incorporation by reference of 08/587,943.

Moreover, the preliminary amendment incorporated by reference 09/046,627 which itself also incorporated by reference 08/587,943; see the discussion of Attachment 13 above.

CHART 3	
Rejected Claim of USP 9021602	Support in **application 10/602,**777 **Attachment 14**
Claims 1, 2, 3, 4, 5, 8, 10, and 12 of USP 9021602	Specification filed 2003-06-25, page 1:12-14 “This application claims the benefit of U.S. patent application Serial No. **08/587,943**, filed January 17, 1996, entitled "Method for Stega-Cipher Protection of Computer Code," the entire disclosure of which is **hereby incorporated by reference**.” Specification filed 2003-06-25, page 7:3-12 “Many approaches to digital watermarking leave detection and decoding control with the implementing party of the digital watermark, not the creator of the work to be protected. A set of secure digital watermark implementations address this fundamental control issue forming the basis of key-based approaches. These are covered by the following patents and pending applications, the entire disclosures of which are **hereby incorporated by reference**: US Patent No. 5,613,004 entitled "Steganographic Method and Device" and its derivative US patent application Serial No. 08/775,216, US patent application Serial No. 08/587,944 entitled "Human Assisted Random Key Generation and Application for Digital Watermark System," US Patent Application Serial No. 08/587,943 entitled "Method for Stega-Cipher Protection of Computer Code," US patent application Serial No. **08/677,435** entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data," and US Patent Application Serial No. 08/772,222 entitled "2-Transform Implementation of Digital Watermarks."

CHART 4: Claims 1, 2, 3, 4, 5, 8, l0, and 12 of USP 9021602 have the same support as in 08/587,94, in application 11/895,388 Attachment 15

Application 11/895,388 both incorporates by reference the disclosure of 08/587,943, and also bodily reintroduces the entirety of the specification of 08/587,943, in addition to properly claiming benefit to 08/587,943.

Attachment 15 includes the documents filed 2007-08-24, which includes the portion of the specification at Attachment 15 pdf pages 3-25; portions of the specification at pdf pages 30-52; portions of the specification at pdf page 64; and portions of the specification at pdf pages 65-78. The file shows that Attachment 15 page 64, and pages 65-78 are part of a preliminary amendment filed with the application. A preliminary amendment filed with an application is part of the original disclosure.

Attachment 15, pdf page 3, [0004], properly claimed to benefit to and incorporated by reference application 08/587,943, stating:

> [0001] This application is a divisional of U.S. Patent Application Serial No. 10/602,777, which is a continuation application of U.S. Patent Application Serial No. 09/046,627 (which issued July 22, 2003, as U.S. Patent No. 6,598,162), **which is a continuation-in-part of** U.S. Patent Application Serial No. **08/587,943**, filed Jan. 17, 1996, (which issued April 28, 1998, as U.S. Patent No. 5,745,943). The entire disclosure of U.S. Patent Application No. 09/046,627 (which issued July 22, 2003, as U.S. Patent No. 6,598,162) and U.S. Patent Application Serial No. **08/587,943**, filed Jan. 17, 1996, (which 'issued April 28, 1998, as U.S. Patent No. 5,745,943) are **hereby incorporated by reference** in their entireties.

Attachment 15, pdf page 30, is an update to [0001] listing issued patent numbers, and retained the proper chain of claims to benefit to and retained the incorporated by reference of application 08/587,943,

Attachment 15, pdf page 38, in [0026], lines 4-17**, also incorporated by reference application 08/587,943**, stating:

> These are covered by the following patents and pending applications, the entire disclosures of which **are hereby incorporated by reference**: U.S. Pat. No. 5,613, 004 entitled "Steganographic Method and Device" and its derivative U.S. patent application Ser. No. 08/775,216 (which issued November 11 1997 as U.S. Patent No. 5 687.236). U.S. patent application Ser. No. 08/587,944 entitled "Human Assisted Random Key Generation and Application for Digital Watermark System[[']]"(which issued October 13 1998 as U.S. Patent No. 5,8224,32) US. patent application Ser. No. **08/587,943** entitled "Method for Stega-Cipher Protection of Computer Code[[,]]//(which issued April 28. 1998 as US. Patent No. 5.748.569) U.S. patent application Ser. No. 08/677,435 entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data[[,]]//(which issued March 30, 1999 as U.S. Patent

NO. 5,889,868) and US. patent application Ser. No. 08/772,222 entitled "Z-Transform Implementation of Digital Watermarks[[.]]"(which issued June 20. 2000 as U.S. Patent No. 6078 664).

Attachment 15, pdf page 64, lines 3-8, **also incorporated by reference application 08/587,943**, stating:

> ...The entire disclosure of U.S. Patent Application No. 09/046,627 (which issued July 22, 2003, as U.S. Patent No. 6,598,162) and U.S. Patent Application Serial No. **08/587,943**, filed Jan. 17, 1996. (which issued April 28, 1998, as U.S. Patent No. 5,745,943 [sic]) are **hereby incorporated by reference** in their entireties.

Moreover, in Attachment 15, pdf page 82, lines 2-4, in the Remarks accompanying the preliminary amendment, Mr. Moskowitz said: "Applicant has **bodily incorporated** U.S. Patent Application Serial No. **08/587,943**, filed January 17, 1996 (which issued as U.S. Patent No. 5,745,569 on April 28, 1998)." Indeed, that is the case. All of the text of the specification of 08/587,943 is bodily incorporated into application 11/895,388, as shown by comparison of Attachments 15 and 1, as summarized below.

Attachment 15, pdf page 65 et seq specifies an insert to the specification at the location after [0002]. This insert has the same text as Attachment 1 (08/587,943), pdf page 8, line 1, to page 12, line 22. This text is the entirety of the disclosure from the beginning of the disclosure to the end of the Background section in Attachment 1 of application 08/587,943.

Attachment 15, pdf page 6 et seq specifies a two paragraph insert to the Summary of the Invention section. The text of this insert is the same as the text of the two paragraph Summary of the Invention section of Attachment 1, which is at Attachment 1, pages 13-14.

Attachment 15, pdf page 6 et seq specifies an insert to the Detailed Description section after [0050]. The text of this insert is the same as the text of the Detailed Description section of Attachment 1, which is at Attachment 1 pdf page 8, line 11 through page 22. And the specification of 08/587,943 end at Attachment 1 pdf page 22.

Thus, Mr. Moskowitz reintroduced, bodily, the entire text of the specification of application 08/587,943, into application 11/895,388.

Chart 4 shows support based upon express disclosure identical to the express disclosure in 08/587,943, application 11/895,388, as filed. Attachment 15, pages 3-25 constitute the clean copy of the disclosure of application 11/895,388, as filed, containing this disclosure. Chart 4 therefore cites exclusively to Attachment 15, pages 3-25.

CHART 4	
Rejected Claim of USP 9021602	Support in application 11/895,388 **Attachment 15** (issued as USP 9104842), pages 3-25

1. A computer based method for accessing functionality provided by an application software comprising:	Attachment 15, pdf page 21, Par [0056], last sentence "The key is necessary to access the underlying code, i.e., what the user understands to be the application program." This statement follows the explanation starting at pdf page 20, in [0054] to [0057], regarding how the software is compiled to encode certain code resources deemed "essential" for the functionality of the software in data resources. Attachment 15, pdf pages 21-22, Par [0057], last sentence to [0060] then explains how a user used the software including the functionality. That is how to perform the method of the preamble of this claim 1.
storing said application software in non transient memory of a computer;	Attachment 15, pdf pages 23-24, [0065] refer to the program being "loaded" which means copied from slow memory (like disc) to fast memory (like RAM). Mr. Moskowitz noted that both of those forms (slow and fast) memory are "non transient." Mr. Moskowitz noted that "non transient" were words the USPTO recommended everyone use in reference to memory when some court decision stated that memory might read on a "signal" and might be considered not patentable subject matter. Mr. Moskowitz noted he was told that the USPTO therefore took a "liberal" view of support for "non transient", basically allowing anyone claiming something stored in memory in a patent application that was not disclosing signals as memory, to add "non transient" to avoid adverse court invalidity determinations.
said application software in said computer prompting a user to enter into said computer personalization information;	Attachment 15, pdf page 4 [0005] notes that it was well known for computer software to prompt a use for information at startup. Attachment 15, pdf page 22, [0058] discloses that the software prompts the user to enter personalization information when run for the first time. Pdf page 22, [0057] states "The application can then operate as follows: 1) when it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. This can include a particular computer configuration;."
said application software storing, in said non transient memory, in a personalization data resource, both computer configuration information of said computer, and a license code	Attachment 15, pdf page 22, the next two lines ([0058] and [0059), to state "2) *it* stores *this* information in a personalization data resource." Mr. Moskowitz in his 131 declaration states that the "it" is the software, and that the "this information" refers to the personalization information which lines 27 and 28 state may include both license code and computer configuration.

entered in response to said prompting;	Mr. Moskowitz also stated in his declaration that the statement that item "2)" follows immediately after item "1)" in the sentence explaining operation of the software indicates that the storing of this information is in response to the entering of the information by the user.
said application software in said computer generating a proper decoding key,	Attachment 15, pdf page 22, [0060] continues "3) Once it has the license code, it can then generate the proper decoding key to access the essential code resources." As Mr. Moskowitz noted in his declaration, this shows generating the proper decoding key.
said generating comprising using said license code; and	Attachment 15, pdf page 22, in the very next paragraph, [0061], states "Note that the application can be copied in an uninhibited manner, but must contain the license code issued to the licensed owner, to access its essential code resources." Mr. Moskowitz notes in his declaration that this indicates that the license code is essential to access the code resource, and accessing requires the decoding key, which indicates that the license code is also essential to generating the decoding key. Mr. Moskowitz also explained the significance of the key and that generating the key requires the license code, as disclose in the assembly of the software, corresponding to Attachment 15, pages 21-22, [0056] - [0057], when describing how the software is assembled. See [0056], lines 18-20 "This method, then, is to choose the key so that it corresponds, is equal to, or is a function of, a license code or license descriptive information." Mr. Moskowitz stated in his declaration that, in other words, this passage disclosed that generating the decode key may require the licence code.
wherein said application software, in said computer, cannot access at least one encoded code resource of said application software, unless said license code is stored in said personalization data resource.	Attachment 15, pdf page 21, [0056], states that "The end result [of compiling the software is that]...these essential code resources... are not accessible at run-time, without the key." Mr. Moskowitz noted in his 131 declaration that this the same thing as stated by this wherein clause in claim 1.

2. The method of claim 1, wherein said encoded code resource is encoded in at least one data resource.	Attachment 15, pdf page 21, [0057], in describing the assembly utility, states that the assembly utility "encodes one or several essential resources into one or several data resources." Attachment 15, pdf page 21, [0056], states that "The utility will chose one or several essential code resources, and encode them into one or several data resources." This discloses that "code resource" are "encoded" and that they are encoded in at least one data resource, as claimed.
3. The method of claim 1 wherein said encoded code resource is steganographically encoded.	Attachment 15, pdf page 6, [0013] explains that "steganography" refers to "hiding something in plain view." Attachment 15, pdf pages4-5, [0008], discloses that the first method "involves hiding necessary 'parts' or code 'resources' ... in digitized sample resources." Attachment 15, pdf pages 4-5, [0008], explains that "It is desirable to use a 'stega-cipher' .... process to hide necessary parts of resource of executable object code in the digitized sample resource." Attachment 15, pdf pages 21, [0056], discloses that the encoding may be "using the stegacipher process." These support use of the adverb "steganographically" when referring to encoding that uses steganography for hiding the code resources in data resources.
4. The method of claim 3 wherein said encoded code resource is encoded in a data resource.	Claim 4 is the same as claim 2, but depends upon claim 3 instead of claim 1. So discussion of claim 2 herein above applies here.
5. The method of claim 1 wherein said computer configuration information is stored in a data resource.	Attachment 15, pdf page 22, [0058], in the description of the operation of the software application, stated "This can include a particular computer configuration; 2) it stores *this information* in a personalization *data resource*." That is, Mr. Moskowitz clearly stated that the computer configuration information may be stored in a data resource.

8. The method of claim 1 wherein said computer comprises a processor and said application software using said processor in said prompting and said storing.	The Attachment 15, disclosure is replete with references to computers in the context of digital computer which necessarily convey the presence of a processor. For example, in discussing the structure of software, Mr. Moskowitz refers to "the instructions" (pdf page 19, [0052]) which immediately conveys a processor for acting on instruction. Similarly, he refers to "executable code" (pdf page 19, [0052]) which immediately conveys a processor for acting on the code. At pdf page 22, [0057] to [0058], in describing a software application including essential code resources encoded in data resource, the specification states "The application can then operate as follows: 1) when it is run for the first time...." Mr. Moskowitz stated that 'running' is a colloquial express for a digital computer executing instructions in a software program. Digital computers necessarily include a processor. Similarly, at pdf pages 23, [0063], Attachment 15 describes a preferred embodiment as implemented in an embedded system with a minimal operating system. Further, at pdf page 24, [0065], Attachment 15 stated that "the present invention concerns itself with any application software that may be used in general computing devices." Mr. Moskowitz noted in his 131 declaration that the term "general computer devices" immediately conveys a processor for use by application software for both prompting (an I/O function) and storing (a data storage function).
10. A computer program product storing in a non transitory storage media computer application software code for an application software product, which, when run by a computer system, causes said computer system to perform the following for accessing functionality provided by said application software product, comprising:	This claim recites a description of software stored on some physical medium. In the Background section in Attachment 15, pdf page 4, [0005] describes that software may be stored on a user's hard drive, when referring to attempts to enforce licencing. That is, in [0005], Attachment 15 states "Further methods include network-based searches of a user's hard drive and comparisons between what is registered to that user and what is actually installed on the user's general computing device." Attachment 15, pdf page 21, [0056] refers to "install[ed] ... copies," which refers to installation on a drive. A computer's drive is a product. Attachment 15, pdf page 5, [0009], refers to storing code in computer memory, stating "It is also desirable to randomly reorganize program memory structure intermittently during program run time." Attachment 15, pdf page 18, [0051], refers to "the order of the machine instructions. .. In the computer memory." Computer memory is a product.

storing said application software code in non transient memory of a computer system; said application software code in said computer system prompting a user to enter into said computer system personalization information; said application software code storing, in said non transient memory, in a personalization data resource, both computer configuration information of said computer system, and a license code entered in response to said prompting; said application software code in said computer system generating a proper decoding key, said generating comprising using said license code; and wherein said application software code, in said computer system, cannot access at least one encoded code resource of said application software code, unless said license code is stored in said personalization data resource.	This is the same recitation appearing in claim 1. See the support for of claim 1 herein above.

12. The product of claim 10 wherein said computer program product causes storing of said encoded code resource in a data resource in non transient memory of said computer.	Claim 12 contains the same recitation ("storing of said encoded code resource in a data resource ") as claim 2. Attachment 15 discloses this feature for the reasons stated for claim 1.

CHART 5: Claims 1, 2, 3, 4, 5, 8, l0, and 12 of USP 9021602 and the same support as in 08/587,943, in application 13/556,420 Attachment 16

Attachment 16 contains the documents filed 2012-07-24 forming application 13/556,420, as filed. These documents include an original specification (at pdf pages 7-28) and a preliminary amendment containing one page of specification (pdf page 4) updating the [0001] paragraph's benefit claim chain information. The remarks in the preliminary amendment state that the amendment "updates the priority [sic; benefit] claim and cancels claims 2-57."

The preliminary amendment to the specification merely amends paragraph [0001] by making a proper benefit claim chain including the prior filed application 11/895,388, and maintaining the incorporation by reference to application 08/587,943, stating (at Attachment 16 page 4):

At Page 1, please replace Paragraph [0001] with the following paragraph:

> [0001] This application is a continuation of U.S. Application No. 11/895,388, filed August 24, 2007, which is a division[[al]] of U.S. patent application [[Ser.]] No. 101602,777, filed June 25, 2003, issued February 16, 2010 as U.S. Patent No. 7,664,263, which is a continuation ~~application~~ of U.S. patent application [[Ser. No.]] 091046,627, filed March 24, 1998, (~~which~~ issued Jul. 22, 2003, as U.S. Pat. No. 6,598,162[[)]], which is a continuation-in-part of U.S. patent application [[Ser.]] No. 08/587,943, filed Jan. 17, 1996, (~~which~~ issued Apr. 28, 1998, as U.S. Pat. No. 5,745,943[[)]]. The entire disclosure of U.S. Application No. 11/895,388, filed August 24,2007, U.S. patent application [[Ser.]] No. 091046,627 (which issued JuI. 22, 2003, as U.S. Pat. No. 6,598,162[[)]] and U.S. patent application [[Ser.]] No. **08/587,943,** filed Jan. 17, 1996, (~~which~~ issued Apr. 28, 1998, as U.S. Pat. No. 5,745,943[[)]] are **hereby incorporated by reference in their entireties**.

The undersigned compared, paragraph by paragraph, the paragraphs [0001] to [0068], forming the specification at pdf pages 7-28 of Attachment 16, to the specification at pdf pages 3-25 of Attachment 15. The undersigned observed that the corresponding numbered paragraphs in both specifications have the same starting and ending words, and appear to contain identical text. That is, pdf pages 7-28 of Attachment 16 appear to be a refiling of pdf pages 3-25 of Attachment 15. Therefore, the specification of application 13/556,420 contains exactly the same disclosure in its paragraphs [0001]-[0068] as does application 11/895,388. Therefore, application 13/556,420 supports claims 1, 2, 3, 4, 5, 8, l0, and 12 of USP 9021602 for the same reasons as application 11/895,388.

To avoid doubt, however, the undersigned copies the claim support chart for application 11/895,388 herein below, and replaces the pdf page number citations (but not paragraph numbers which are the same) from that chart with the pdf page numbers of Attachment 16 where the cited paragraphs appear in Attachment 16.

CHART 5	
Rejected Claim of USP 9021602	Support in **application 11/895,388 Attachment 15** (issued as USP 9104842), pages 3-25
1. A computer based method for accessing functionality provided by an application software comprising:	Attachment 16, pdf page 24, Par [0056], last sentence "The key is necessary to access the underlying code, i.e., what the user understands to be the application program." This statement follows the explanation starting at pdf page 23, in [0054] to [0057], regarding how the software is compiled to encode certain code resources deemed "essential" for the functionality of the software in data resources. Attachment 16, pdf page 25, Par [0057], last sentence to [0060] then explains how a user used the software including the functionality. That is how to perform the method of the preamble of this claim 1.
storing said application software in non transient memory of a computer;	Attachment 16, pdf pages 27, [0065] refer to the program being "loaded" which means copied from slow memory (like disc) to fast memory (like RAM). Mr. Moskowitz noted that both of those forms (slow and fast) memory are "non transient." Mr. Moskowitz noted that "non transient" were words the USPTO recommended everyone use in reference to memory when some court decision stated that memory might read on a "signal" and might be considered not patentable subject matter. Mr. Moskowitz noted he was told that the USPTO therefore took a "liberal" view of support for "non transient", basically allowing anyone claiming something stored in memory in a patent application that was not disclosing signals as memory, to add "non transient" to avoid adverse court invalidity determinations.
said application software in said computer prompting a user to enter into said computer personalization information;	Attachment 16, pdf pages7-8 [0005] notes that it was well known for computer software to prompt a use for information at startup. Attachment 16, pdf page 25, [0058] discloses that the software prompts the user to enter personalization information when run for the first time. Pdf page 25, [0057] states "The application can then operate as follows: 1) when it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. This can include a particular computer configuration;."

said application software storing, in said non transient memory, in a personalization data resource, both computer configuration information of said computer, and a license code	Attachment 16, pdf page 25, the next two lines ([0058] and [0059), to state "2) *it* stores *this* information in a personalization data resource." Mr. Moskowitz in his 131 declaration states that the "it" is the software, and that the "this information" refers to the personalization information which lines 27 and 28 state may include both license code and computer configuration.
entered in response to said prompting;	Mr. Moskowitz also stated in his declaration that the statement that item "2)" follows immediately after item "1)" in the sentence explaining operation of the software indicates that the storing of this information is in response to the entering of the information by the user.
said application software in said computer generating a proper decoding key,	Attachment 16, pdf page 25, [0060] continues "3) Once it has the license code, it can then generate the proper decoding key to access the essential code resources." As Mr. Moskowitz noted in his declaration, this shows generating the proper decoding key.
said generating comprising using said license code; and	Attachment 16, pdf page 25, in the very next paragraph, [0061], states "Note that the application can be copied in an uninhibited manner, but must contain the license code issued to the licensed owner, to access its essential code resources." Mr. Moskowitz notes in his declaration that this indicates that the license code is essential to access the code resource, and accessing requires the decoding key, which indicates that the license code is also essential to generating the decoding key. Mr. Moskowitz also explained the significance of the key and that generating the key requires the license code, as disclose in the assembly of the software, corresponding to Attachment 16, pages 24-25, [0056] - [0057], when describing how the software is assembled. See [0056], lines 18-20 "This method, then, is to choose the key so that it corresponds, is equal to, or is a function of, a license code or license descriptive information." Mr. Moskowitz stated in his declaration that, in other words, this passage disclosed that generating the decode key may require the licence code.

wherein said application software, in said computer, cannot access at least one encoded code resource of said application software, unless said license code is stored in said personalization data resource.	Attachment 16, pdf pages 24-25, [0056], states that “The end result [of compiling the software is that]...these essential code resources... are not accessible at run-time, without the key.” Mr. Moskowitz noted in his 131 declaration that this the same thing as stated by this wherein clause in claim 1.
2. The method of claim 1, wherein said encoded code resource is encoded in at least one data resource.	Attachment 16, pdf page 25, [0057], in describing the assembly utility, states that the assembly utility “encodes one or several essential resources into one or several data resources.” Attachment 16, pdf pages 24-25, [0056], states that “The utility will chose one or several essential code resources, and encode them into one or several data resources.” This discloses that “code resource” are “encoded” and that they are encoded in at least one data resource, as claimed.
3. The method of claim 1 wherein said encoded code resource is steganographically encoded.	Attachment 16, pdf page 10, [0013] explains that “steganography” refers to “hiding something in plain view.” Attachment 16, pdf pages 8-9, [0008], discloses that the first method “involves hiding necessary ‘parts’ or code ‘resources’ ... in digitized sample resources.” Attachment 16, pdf pages 8-9, [0008], explains that “It is desirable to use a ‘stega-cipher’ .... process to hide necessary parts of resource of executable object code in the digitized sample resource.” Attachment 16, pdf pages 24-25, [0056], discloses that the encoding may be “using the stegacipher process.” These support use of the adverb “steganographically” when referring to encoding that uses steganography for hiding the code resources in data resources.
4. The method of claim 3 wherein said encoded code resource is encoded in a data resource.	Claim 4 is the same as claim 2, but depends upon claim 3 instead of claim 1. So discussion of claim 2 herein above applies here.

5. The method of claim 1 wherein said computer configuration information is stored in a data resource.	Attachment 16, pdf page 25, [0058], in the description of the operation of the software application, stated "This can include a particular computer configuration; 2) it stores *this information* in a personalization *data resource*." That is, Mr. Moskowitz clearly stated that the computer configuration information may be stored in a data resource.
8. The method of claim 1 wherein said computer comprises a processor and said application software using said processor in said prompting and said storing.	The Attachment 16, disclosure is replete with references to computers in the context of digital computer which necessarily convey the presence of a processor. For example, in discussing the structure of software, Mr. Moskowitz refers to "the instructions" (pdf page 22, [0052]) which immediately conveys a processor for acting on instruction. Similarly, he refers to "executable code" (pdf page 22, [0052]) which immediately conveys a processor for acting on the code. At pdf page 25, [0057] to [0058], in describing a software application including essential code resources encoded in data resource, the specification states "The application can then operate as follows: 1) when it is run for the first time...." Mr. Moskowitz stated that 'running' is a colloquial express for a digital computer executing instructions in a software program. Digital computers necessarily include a processor. Similarly, at pdf pages 26, [0063], Attachment 16 describes a preferred embodiment as implemented in an embedded system with a minimal operating system. Further, at pdf page 27, [0065], Attachment 16 stated that "the present invention concerns itself with any application software that may be used in general computing devices." Mr. Moskowitz noted in his 131 declaration that the term "general computer devices" immediately conveys a processor for use by application software for both prompting (an I/O function) and storing (a data storage function).

10. A computer program product storing in a non transitory storage media computer application software code for an application software product, which, when run by a computer system, causes said computer system to perform the following for accessing functionality provided by said application software product, comprising:	This claim recites a description of software stored on some physical medium. In the Background section in Attachment 16, pdf page 7, [0005] describes that software may be stored on a user's hard drive, when referring to attempts to enforce licencing. That is, in [0005], Attachment 16 states "Further methods include network-based searches of a user's hard drive and comparisons between what is registered to that user and what is actually installed on the user's general computing device." Attachment 16, pdf page 24, [0056] refers to "install[ed] ... copies," which refers to installation on a drive. A computer's drive is a product. Attachment 16, pdf page 9, [0009], refers to storing code in computer memory, stating "It is also desirable to randomly reorganize program memory structure intermittently during program run time." Attachment 16, pdf page 21-22, [0051], refers to "the order of the machine instructions. .. In the computer memory." Computer memory is a product.

storing said application software code in non transient memory of a computer system; said application software code in said computer system prompting a user to enter into said computer system personalization information; said application software code storing, in said non transient memory, in a personalization data resource, both computer configuration information of said computer system, and a license code entered in response to said prompting; said application software code in said computer system generating a proper decoding key, said generating comprising using said license code; and wherein said application software code, in said computer system, cannot access at least one encoded code resource of said application software code, unless said license code is stored in said personalization data resource.	This is the same recitation appearing in claim 1. See the support for of claim 1 herein above.

12. The product of claim 10 wherein said computer program product causes storing of said encoded code resource in a data resource in non transient memory of said computer.	Claim 12 contains the same recitation ("storing of said encoded code resource in a data resource ") as claim 2. Attachment 16 discloses this feature for the reasons stated for claim 1.

CHART 6: Claims 1, 2, 3, 4, 5, 8, l0, and 12 of USP 9021602 have the same support as in 08/587,943, in USP 9021602.

USP 9021602 contains the same support for Claims 1, 2, 3, 4, 5, 8, l0, and 12 USP 9021602 as does application 08/587,943, Attachment 1, both by incorporation and expressly. As to incorporation:

col. 1:13-20 states:

> The entire disclosure of U.S. application Ser. No. 13/556,420, filed Jul. 24, 2012, U.S. application Ser. No. 111895,388, filed Aug. 24, 2007, U.S. patent application Ser. No. 09/046,627, issued Jul. 22, 2003, as U.S. Pat. No. 6,598,162, and U.S. patent **application Ser. No. 08/587,943**, filed Jan. 17, 1996, issued Apr. 28,1998, as U.S. Pat. No.5, 745,569 are **hereby incorporated by reference** in their entireties.

And col. 6:14-36 states:

> ...These are covered by the following patents and pending applications, the entire disclosures of which are **hereby incorporated by reference**: U.S. Pat. No. 5,613,004 entitled "Steganographic Method and Device" and its derivative u.s. patent application Ser. No. 081775,216 (which 20 issued Nov. 11, 1997, as U.S. Pat. No. 5,687,236), U.S. patent application Ser. No. 08/587,944 entitled "Human Assisted Random Key Generation and Application for Digital Watermark System" (which issued Oct. 13, 1998, as U.S. Pat. No. 5,822,432), U.S. patent **application Ser. No. 08/587,943** entitled "Method for Stega-Cipher Protection of Computer Code" (which issued Apr. 28,1998, as U.S. Pat. No. 5,748, 569), U.S. patent application Ser. No. 08/677,435 entitled "Optimization Methods for the Insertion, Protection, and Detection of Digital Watermarks in Digitized Data" (which issued Mar. 30, 1999, as U.S. Pat. No. 5,889,868) and U.S. patent application Ser. No. 081772,222 entitled "Z-Transfonn Implementation of Digital Watennarks" (which issued Jun. 20, 2000, as U.S. Pat. No. 6,078,664). Public key cryptosystems are described in U.S. Pat. Nos. 4,200,770, 4,218,582, 4,405,829 and 4,424,414, the entire disclosures of which are also hereby incorporated by reference.

Rejected Claim of USP 9021602	Support in USP 9021602, Exhibit 1

1. A computer based method for accessing functionality provided by an application software comprising:	USP 9021602 Col. 13:42-44 "The key is necessary to access the underlying code, i.e., what the user understands to be the application program." This statement follows the explanation at col. 13:9-42 regarding how the software is compiled to encode certain code resources deemed "essential" for the functionality of the software in data resources. Col 13:58-67 then explains how a user used the software including the functionality. That is how to perform the method of the preamble of this claim 1.
storing said application software in non transient memory of a computer;	USP 9021602 Col. 12: 37 refer to the program being "loaded" which means copied from slow memory (like disc) to fast memory (like RAM). Both of those forms (slow and fast) memory are "non transient."
said application software in said computer prompting a user to enter into said computer personalization information;	USP 9021602 col. 1: 45-48 points out that it was well known for computer software to prompt for information at startup. Col. 13:58-61 discloses that the software prompts the user to enter personalization information when run for the first time. Col. 13:58-61 states "The application can then operate as follows: 1) when it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. This can include a particular computer configuration;."
said application software storing, in said non transient memory, in a personalization data resource, both computer configuration information of said computer, and a license code	Col. 13:64-65 continues the same sentence, stating "2) *it* stores *this* information in a personalization data resource;" The "it" is the software. The "this information" refers to the personalization information which lines 27 and 28 state may include both license code and computer configuration.
entered in response to said prompting;	That statement that item "2)" follows immediately after item "1)" in the sentence explaining operation of the software indicates that the storing of this information is in response to the entering of the information by the user.
said application software in said computer generating a proper decoding key,	Col. 13: 66-67 states "3) Once it has the license code, it can then generate the proper decoding key to access the essential code resources." This shows generating the proper decoding key.

said generating comprising using said license code; and	The very next paragraph, col. 14: 1-6, states “Note that the application can be copied in an uninhibited manner, but must contain the license code issued to the licensed owner, to access its essential code resources.” This indicates that the license code is essential to access the code resource, and accessing requires the decoding key, which indicates that the license code is also essential to generating the decoding key. Mr. Moskowitz explained the significance of the key and that generating the key requires the license code, back in col. 13:37-44 when describing how the software is assembled. “This method, then, is to choose the key so that it corresponds, is equal to, or is a function of, a license code or license descriptive information.” In other words, Mr. Moskowitz disclosed that generating the decode key may requires the licence code.
wherein said application software, in said computer, cannot access at least one encoded code resource of said application software, unless said license code is stored in said personalization data resource.	Col. 13:18-21 states that “The end result [of compiling the software is that]...these essential code resources... are not accessible at run-time, without the key.” This the same thing as stated by this wherein clause in claim 1.
2. The method of claim 1, wherein said encoded code resource is encoded in at least one data resource.	In describing the assembly utility, at col. 13:49-50, USP 9021602 states that the assembly utility “encodes one or several essential resources into one or several data resources.” At col. 13:14-18, USP 9021602 states that “The utility will chose one or several essential code resources, and encode them into one or several data resources.” This discloses that “code resource” are “encoded” and that they are encoded in at least one data resource, as claimed.

3. The method of claim 1 wherein said encoded code resource is steganographically encoded.	At col. 3:21-23, USP 9021602 explains that “steganography” refers to “hiding something in plain view.” At col. 12:25-28, USP 9021602 discloses that the first method of the invention “involves hiding necessary ‘parts’ or code ‘resources’ in digitized sample resources.” At col. 2:21-23, USP 9021602 explains that “It is desirable to use a ‘stega-cipher’ .... process to hide necessary parts of resource of executable object code in the digitized sample resource.” At col. 13:14-17, USP 9021602 discloses that the encoding may be “using the stegacipher process.” Mr. Moskowitz believes these support use of the adverb “steganographically” when referring to encoding that uses steganography for hiding the code resources in data resources.
4. The method of claim 3 wherein said encoded code resource is encoded in a data resource.	Claim 4 is the same as claim 2, but depends upon claim 3 instead of claim 1. So the discussion of claim 2 applies here.
5. The method of claim 1 wherein said computer configuration information is stored in a data resource.	The operation of the software application at col. 13:62-63 stated “This can include a particular computer configuration; 2) it stores *this information* in a personalization *data resource*.” That clearly stated that the computer configuration information may be stored in a data resource.

8. The method of claim 1 wherein said computer comprises a processor and said application software using said processor in said prompting and said storing.	USP 9021602 is replete with references to computers in the context of digital computer which necessarily convey the presence of a processor. For example, in discussing the structure of software, USP 9021602 refers to "the instructions" (col. 11:58) which immediately conveys a processor for acting on instruction. Similarly, USP 9021602 refers to "executable code" (col. 12:1) which immediately conveys a processor for acting on the code. At col. 13:58-60, in describing a software application including essential code resources encoded in data resource, USP 9021602 states "The application can then operate as follows: 1) when it is run for the first time...." Running is a colloquial expression for a digital computer executing instructions in a software program. Digital computers necessarily include a processor. Similarly, col. 14:35-37 describes a preferred embodiment as implemented in an embedded system with a minimal operating system. Further, at col. 15:27-29 states that "the present invention concerns itself with any application software that may be used in general computing devices." Mr. Moskowitz in his 131 declaration note that the term "general computer devices" immediately conveys a processor for use by application software for both prompting (an I/O function) and storing (a data storage function).
10. A computer program product storing in a non transitory storage media computer application software code for an application software product, which, when run by a computer system, causes said computer system to perform the following for accessing functionality provided by said application software product, comprising:	This is a description of software stored on some physical medium. The Field of the Invention section describes that software may be stored on a user's hard drive, when referring to attempts to enforce licencing. That is, at col. 1:54-57 states "Further methods include network-based searches of a user's hard drive and comparisons between what is registered to that user and what is actually installed on the user's general computing device." At col. 13:35-36 refers to "install[ed] ... copies," which refers to installation on a drive. A computer's drive is a product. Col. 2:41-46 refers to storing code in computer memory, stating "It is also desirable to randomly reorganize program memory structure intermittently during program run time." Col. 11:40-44 refers to "the order of the machine instructions. .. In the computer memory." Computer memory is a product.

storing said application software code in non transient memory of a computer system; said application software code in said computer system prompting a user to enter into said computer system personalization information; said application software code storing, in said non transient memory, in a personalization data resource, both computer configuration information of said computer system, and a license code entered in response to said prompting; said application software code in said computer system generating a proper decoding key, said generating comprising using said license code; and wherein said application software code, in said computer system, cannot access at least one encoded code resource of said application software code, unless said license code is stored in said personalization data resource.	This is the same recitation appearing in claim 1. See the discussion of claim 1 herein above.

12. The product of claim 10 wherein said computer program product causes storing of said encoded code resource in a data resource in non transient memory of said computer.	Claim 12 contains the same recitation ("storing of said encoded code resource in a data resource ") as claim 2. USP 9021602 discloses this feature for the reasons stated for claim 1.

/Richard Neifeld/
RICHARD NEIFELD
Attorney of record, Registration No. 35,299

PATENT ASSIGNMENT COVER SHEET

Electronic Version v1.1
Stylesheet Version v1.2

EPAS ID: PAT3486466

SUBMISSION TYPE:	NEW ASSIGNMENT
NATURE OF CONVEYANCE:	ASSIGNMENT

CONVEYING PARTY DATA

Name	Execution Date
SCOTT A. MOSKOWITZ	08/14/2015

RECEIVING PARTY DATA

Name:	WISTARIA TRADING LTD
Street Address:	CLARENDON HOUSE, 2 CHURCH STREET
City:	HAMILTON
State/Country:	BERMUDA
Postal Code:	HM 11

PROPERTY NUMBERS Total: 40

Property Type	Number
Patent Number:	6078664
Patent Number:	6598162
Patent Number:	6205249
Patent Number:	7035409
Application Number:	09767733
Patent Number:	7664263
Application Number:	10805484
Patent Number:	7738659
Application Number:	11895388
Patent Number:	8265276
Patent Number:	8542831
Patent Number:	8930719
Patent Number:	9021602
Application Number:	13937106
Application Number:	14258118
Application Number:	14258171
Application Number:	14258237
Application Number:	14542712
PCT Number:	US1999007262
Patent Number:	7287275

Property Type	Number
Patent Number:	8224705
Patent Number:	7530102
Patent Number:	8104079
Patent Number:	8473746
Patent Number:	8706570
Patent Number:	RE44222
Patent Number:	RE44307
Application Number:	60213489
Application Number:	60147134
Application Number:	60125990
Application Number:	13970574
Application Number:	60372788
Application Number:	14666754
Application Number:	60234199
Application Number:	60169274
Application Number:	61794141
Application Number:	61952823
Application Number:	61953684
PCT Number:	US2000033126
PCT Number:	US2000018411

CORRESPONDENCE DATA

Fax Number: (703)415-0013

Correspondence will be sent to the e-mail address first; if that is unsuccessful, it will be sent using a fax number, if provided; if that is unsuccessful, it will be sent via US Mail.

Phone:	7034150012
Email:	bmargulies@neifeld.com
Correspondent Name:	BRUCE T. MARGULIES
Address Line 1:	4813-B EISENHOWER AVE
Address Line 4:	ALEXANDRIA, VIRGINIA 22304

ATTORNEY DOCKET NUMBER:	SCOT0001
NAME OF SUBMITTER:	BRUCE T. MARGULIES
SIGNATURE:	/BruceMargulies/
DATE SIGNED:	08/17/2015

Total Attachments: 6

source=ExecutedAssignment_SM_Assignor#page1.tif
source=ExecutedAssignment_SM_Assignor#page2.tif
source=ExecutedAssignment_SM_Assignor#page3.tif
source=ExecutedAssignment_SM_Assignor#page4.tif
source=ExecutedAssignment_SM_Assignor#page5.tif

NEIFELD REF:	SCOT0001
CLIENT REF:	SCOT0001

37 CFR 1.46 ASSIGNMENT PATENTS AND APPLICATIONS

WHEREAS, the assignor entity (or entities) and their principle place of business and state of incorporation, listed below (hereinafter referred to as "ASSIGNOR"):

FIRST ASSIGNOR ENTITY:

NAME	SCOTT A. MOSKOWITZ
ADDRESS (in order: street; city; state; country; postal code.)	1314 E. Las Olas Blvd., #123, Fort Lauderdale, FL 33301
STATE OF INCORPORATION	

own rights in the following applications and patents:

APPLI-CATION NUMBER	FILING DATE	PATENT NUMBER	ISSUE DATE	COUNTRY	Neifeld Docket/TITLE
08772222	12/20/96	6078664	6/20/00	US	SCOT0023-1 Z-transform implementation of digital watermarks
09046627	3/24/98	6598162	7/22/03	US	SCOT0014-1 Method for combining transfer functions with predetermined key creation
09053628	4/2/98	6205249	3/20/01	US	SCOT0019-1 Multiple transform utilization and applications for secure digital watermarking
09644098	8/23/00	7035409	4/25/06	US	SCOT0019-2 Multiple transform utilization and applications for secure digital watermarking
09767733	1/24/01	Abandoned		US	Multiple transform utilization and applications for secure digital watermarking
10602777	6/25/03	7664263	2/16/10	US	SCOT0014-2 Method for combining transfer functions with predetermined key creation

10805484	3/22/04	Abandoned		US	Method and device for monitoring and analyzing signals
11358874	2/21/06	7738659	6/15/10	US	SCOT0019-3 Multiple transform utilization and applications for secure digital watermarking
11895388	8/24/07	Pending		US	SCOT0014-4 Data protection method and device
12655002	12/22/09	8265276	9/11/12	US	SCOT0014-5 Method for combining transfer functions with predetermined key creation
12799894	5/4/10	8542831	9/24/13	US	SCOT0019-4 Multiple transform utilization and application for secure digital watermarking
13556420	7/24/12	8930719	1/6/15	US	SCOT0014-6 Data protection method and device
13794584	3/11/13	9021602	4/28/2015	US	SCOT0014-7 Data protection method and device
13937106	7/8/13	Pending		US	SCOT0019-5 Multiple transform utilization and application for secure digital watermarking
14258118	4/22/14	Pending		US	SCOT0019-8 Multiple transform utilization and application for secure digital watermarking
14258171	4/22/14	Pending		US	SCOT0019-6 Multiple transform utilization and application for secure digital watermarking
14258237	4/22/14	Pending		US	SCOT0019-7 Multiple transform utilization and application for secure digital watermarking
14542712	11/17/14	Pending		US	SCOT0014-8 Data protection method and device
PCTUS9907262	4/2/99	Expired		PCT	Multiple transform utilization and applications for secure digital watermarking
60213489	6/23/2000	Expired		US	SCOT0016-P1 A Secure Personal Content Server

60147134	8/4/1999	Expired		US	SCOT0016-P2 A Secure Personal Content Server
60125990	3/24/1999	Expired		US	SCOT0021-PR UTILIZING DATA REDUCTION IN STEGANOGRAPHIC AND CRYPTOGRAPHIC SYSTEMS
10417231	4/17/2003	7287275	10/23/2007	US	SCOT0018-1 Methods, systems and devices for packet watermarking and efficient provisioning of bandwidth
11900065	9/10/2007	8224705	7/17/2012	US	SCOT0018-2 Methods, systems and devices for packet watermarking and efficient provisioning of bandwidth
11900066	9/10/2007	7530102	5/5/2009	US	SCOT0018-3 Methods, systems and devices for packet watermarking and efficient provisioning of bandwidth
12383289	3/23/2009	8104079	1/24/2012	US	SCOT0018-4 Methods, systems and devices for packet watermarking and efficient provisioning of bandwidth
13273930	10/14/2011	8473746	6/25/2013	US	SCOT0018-5 Methods, systems and devices for packet watermarking and efficient provisioning of bandwidth
13551097	7/17/2012	8706570	4/22/2014	US	SCOT0018-6 Methods, systems and devices for packet watermarking and efficient provisioning of bandwidth
13488357	6/4/2012	RE44222	5/14/2013	US	SCOT0018-7 Methods, systems and devices for packet watermarking and efficient provisioning of bandwidth
13488395	6/4/2012	RE44307	6/18/2013	US	SCOT0018-8 Methods, systems and devices for packet watermarking and efficient provisioning of bandwidth
13970574	8/19/2013	Pending		US	SCOT0018-9 Methods, systems and devices for packet watermarking and efficient provisioning of bandwidth
60372788	4/17/2002	Expired		US	SCOT0018-P1
14666754	3/24/2015	Pending		US	SCOT0020-5 Systems, Methods and Devices for Trusted Transactions

60234199	9/20/2000	Expired		US	SCOT0020-PR1 Improved Security Based on Subliminal and Supraliminal Channels For Data Objects
60169274	12/7/1999	Expired		US	SCOT0020-PR2 Systems, Methods And Devices For Trusted Transactions
60234199	9/20/2000	Expired		US	SCOT0024-PR Improved Security Based on Subliminal and Supraliminal Channels for Data Objects
61794141	3/15/2013	Expired		US	SCOT0025-1 METHODS, SYSTEMS, AND DEVICES FOR GOOD OBFUSCATION AND PLAUSIBLE DENIABILITY
61952823	3/13/2014	Expired		US	SCOT0025-2 METHODS, SYSTEMS, AND DEVICES FOR GOOD OBFUSCATION AND PLAUSIBLE DENIABILITY
61953684	3/14/2014	Expired		US	SCOT0025-3 METHODS, SYSTEMS, AND DEVICES FOR GOOD OBFUSCATION AND PLAUSIBLE DENIABILITY
PCTUS0033126	12/07/2000	Expired		PCT	SYSTEMS, METHODS AND DEVICES FOR TRUSTED TRANSACTIONS
PCT/US2000/018411	7/5/2000	Expired		PCT	COPY PROTECTION OF DIGITAL DATA COMBINING STEGANOGRAPHIC AND CRYPTOGRAPHIC TECHNIQUES

WHEREAS, the assignee entity (or entities), their principle places of business and their state of incorporation listed below (hereinafter referred to as "ASSIGNEE") :

FIRST ASSIGNEE ENTITY:

NAME	Wistaria Trading Ltd
ADDRESS (in order: street; city; state; country; postal code.)	Clarendon House, 2 Church Street, Hamilton HM 11, Bermuda
STATE OF INCORPORATION	

are desirous of acquiring the entire right, title and interest in and to said applications and patents and inventions disclosed or claimed therein and in and to any Letters Patent that may be granted therefore in the United States and its territorial possessions and in any and all foreign countries;

ASSIGNOR, "SCOTT A. MOSKOWITZ", is listed as the inventor in assignments recorded in the USPTO with clerical variations in the inventor's name, such as: "SCOTT A MOSKOWITZ", "SCOTT A MOSKOWITZ", and "SCOTT MOSKOWITZ".

NOW, THEREFORE, in consideration of the sum of FIVE DOLLARS ($5.00), the receipt whereof is hereby acknowledged, and for other good and valuable consideration, ASSIGNOR, by these presents do, at this time, sell, assign and transfer unto said ASSIGNEE the all rights to the said applications and patents, which includes all rights to claim any invention disclosed in any of said applications and patents, in the United States and its territorial possessions and in all foreign countries, and the entire right, title and interest in and to any and all Letters Patent which may be granted in the future or were granted in the past therefor in the United States and its territorial possessions and in any and all foreign countries and in and to any and all divisions, reissues, continuations, substitutions and renewals thereof which may be granted in the future or were granted in the past. This transfer includes all rights to collect for money for and obtain injunctions based upon, past infringement.

ASSIGNOR hereby authorize and request the Patent Office Officials in the United States and its territorial possessions and any and all foreign countries to issue any and all of said Letters Patent, when granted, to said ASSIGNEE as the assignee of ASSIGNOR'S entire right, title and interest in and to the same, for the sole use and behoof of said ASSIGNEE, ASSIGNEE'S successors and assigns, to the full end of the term for which said Letters Patent may be granted, as fully and entirely as the same would have been held by ASSIGNOR had this Assignment and sale not been made.

Further, ASSIGNOR agrees that ASSIGNOR will communicate to said ASSIGNEE or ASSIGNEE'S representatives any facts known to ASSIGNOR respecting said invention, and testify in any legal proceeding, sign all lawful papers, execute all cause any and all of said Letter Patent to be issued to said ASSIGNEE, make all rightful oaths, and, generally do everything possible to aid said ASSIGNEE, and said ASSIGNEE'S successors and assigns, to obtain and enforce protection for said invention in the United States and its territorial possessions and in any and all foreign countries.

The undersigned hereby grants(s) the firm of Neifeld IP Law, P.C. the power to insert on this assignment any further identification, including firm reference number, filing date, execution date, and any other information which may be necessary or desirable in order to comply with the rules of the United States Patent and Trademark Office for recordation of this document.

ASSIGNOR SIGNATURE(S)

FIRST ASSIGNOR ENTITY:

SIGNATURE:	
PRINTED NAME:	SCOTT A. MOSKOWITZ
LEGAL ENTITY NAME	SCOTT A. MOSKOWITZ, an Individual
TITLE AT LEGAL ENTITY:	Individual
DATE SIGNED:	August 14, 2015
AUTHORIZATION:	I am authorized to act on behalf of this entity.

ASSIGNEE SIGNATURES

FIRST ASSIGNEE ENTITY:

SIGNATURE:	
PRINTED NAME:	SCOTT A. MOSKOWITZ
LEGAL ENTITY NAME	Wistaria Trading Ltd
TITLE AT LEGAL ENTITY:	Director
DATE SIGNED:	August 14, 2015
AUTHORIZATION:	I am authorized to act on behalf of this entity.

BTM
Printed: August 10, 2015 (8:04PM)
Y:\FirmForms\Forms_Patent\US\PatentAssignmentOfPatentsAndApplications_WordPerfect.wpd

US008930719B2

(12) **United States Patent**
Moskowitz

(10) **Patent No.: US 8,930,719 B2**
(45) **Date of Patent: Jan. 6, 2015**

(54) **DATA PROTECTION METHOD AND DEVICE**

(76) Inventor: **Scott A. Moskowitz**, Sunny Isles Beach, FL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **13/556,420**

(22) Filed: **Jul. 24, 2012**

(65) **Prior Publication Data**

US 2013/0014271 A1 Jan. 10, 2013

Related U.S. Application Data

(60) Continuation of application No. 11/895,388, filed on Aug. 24, 2007, which is a division of application No. 10/602,777, filed on Jun. 25, 2003, now Pat. No. 7,664,263, which is a continuation of application No. 09/046,627, filed on Mar. 24, 1998, now Pat. No. 6,598,162.

(51) **Int. Cl.**

G06F 21/00	(2013.01)
G06F 21/10	(2013.01)
G06F 21/12	(2013.01)
G06F 21/16	(2013.01)
G06F 21/33	(2013.01)
G06T 1/00	(2006.01)
H04L 9/06	(2006.01)
H04L 9/32	(2006.01)

(52) **U.S. Cl.**

CPC ***G06F 21/10*** (2013.01); ***G06F 21/125*** (2013.01); ***G06F 21/16*** (2013.01); ***G06F 21/335*** (2013.01); ***G06T 1/0021*** (2013.01); ***H04L 9/065*** (2013.01); ***H04L 9/3236*** (2013.01); ***H04L 9/3247*** (2013.01); *G06F 2211/007* (2013.01); *G06F 2221/0737* (2013.01); *G06F 2221/2107* (2013.01); *G06T 2201/0064* (2013.01); *G06T 2201/0083* (2013.01); *H04L 2209/605* (2013.01); *H04L 2209/608* (2013.01)

USPC **713/193**

(58) **Field of Classification Search**

CPC G06F 21/10; G06F 21/335; G06F 21/125; G06F 21/16; G06F 2221/2107; G06F 2211/007; G06F 2221/0737; H04L 9/3247; H04L 9/3236; H04L 9/065

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,947,825	A	3/1976	Cassada
3,984,624	A	10/1976	Waggener
3,986,624	A	10/1976	Cates, Jr. et al.
4,038,596	A	7/1977	Lee
4,200,770	A	4/1980	Hellman et al.

(Continued)

FOREIGN PATENT DOCUMENTS

EP	0372601	6/1990
EP	0565947	10/1993

(Continued)

OTHER PUBLICATIONS

1997, Merriam-Webster's Collegiate Dictionary, 10th Ed., Merriam Webster, Inc., p. 207.

(Continued)

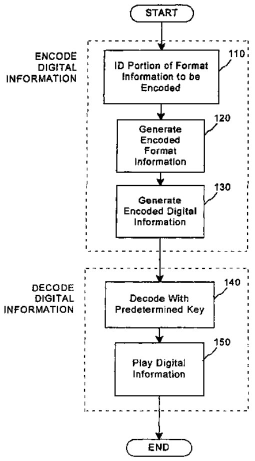
Primary Examiner — Izunna Okeke

(74) *Attorney, Agent, or Firm* — Neifeld IP Law, PC

(57) **ABSTRACT**

An apparatus and method for encoding and decoding additional information into a digital information in an integral manner. More particularly, the invention relates to a method and device for data protection.

50 Claims, 1 Drawing Sheet


START
ENCODE DIGITAL INFORMATION
ID Portion of Format Information to be Encoded
110
Generate Encoded Format Information
120
Generate Encoded Digital Information
130
DECODE DIGITAL INFORMATION
Decode With Predetermined Key
140
Play Digital Information
150
END

Attachment 19 Page 1 of 1

US009104842B2

(12) **United States Patent**
Moskowitz

(10) **Patent No.: US 9,104,842 B2**
(45) **Date of Patent: Aug. 11, 2015**

(54) **DATA PROTECTION METHOD AND DEVICE**

(76) Inventor: **Scott A. Moskowitz**, Sunny Isles Beach, FL (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1965 days.

(21) Appl. No.: **11/895,388**

(22) Filed: **Aug. 24, 2007**

(65) **Prior Publication Data**

US 2008/0016365 A1 Jan. 17, 2008

Related U.S. Application Data

(60) Division of application No. 10/602,777, filed on Jun. 25, 2003, now Pat. No. 7,664,263, which is a continuation of application No. 09/046,627, filed on Mar. 24, 1998, now Pat. No. 6,598,162.

(51) **Int. Cl.**

G06F 21/16	(2013.01)
G06F 21/10	(2013.01)
G06F 21/12	(2013.01)
G06F 21/33	(2013.01)
G06T 1/00	(2006.01)
H04L 9/06	(2006.01)
H04L 9/32	(2006.01)

(52) **U.S. Cl.**

CPC ***G06F 21/10*** (2013.01); ***G06F 21/125*** (2013.01); ***G06F 21/16*** (2013.01); ***G06F 21/335*** (2013.01); ***G06T 1/0021*** (2013.01); ***H04L 9/065*** (2013.01); ***H04L 9/3236*** (2013.01); ***H04L 9/3247*** (2013.01); *G06F 2211/007* (2013.01); *G06F 2221/0737* (2013.01); *G06F 2221/2107* (2013.01); *G06T 2201/0064* (2013.01); *G06T 2201/0083* (2013.01); *H04L 2209/605* (2013.01); *H04L 2209/608* (2013.01)

(58) **Field of Classification Search**

CPC H04L 63/0428; H04L 2209/608; H04L 2209/60

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,947,825	A	3/1976	Cassada
3,984,624	A	10/1976	Waggener
3,986,624	A	10/1976	Cates, Jr. et al.
4,038,596	A	7/1977	Lee
4,200,770	A	4/1980	Hellman et al.
4,218,582	A	8/1980	Hellman et al.
4,339,134	A	7/1982	Macheel
4,390,898	A	6/1983	Bond et al.
4,405,829	A	9/1983	Rivest et al.
4,424,414	A	1/1984	Hellman et al.
4,528,588	A	7/1985	Lofberg

(Continued)

FOREIGN PATENT DOCUMENTS

EP	0372601	6/1990
EP	0372601 A1	6/1990

(Continued)

OTHER PUBLICATIONS

U.S. Appl. No. 08/999,766, filed Jul. 23, 1997, entitled "Steganographic Method and Device".

(Continued)

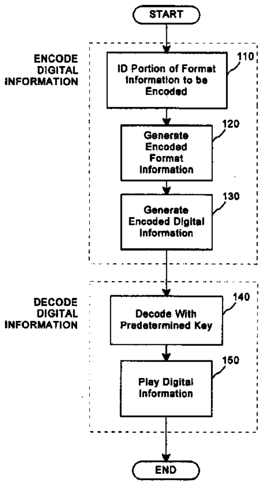
Primary Examiner — Izunna Okeke

(74) *Attorney, Agent, or Firm* — Neifeld IP Law, PC

(57) **ABSTRACT**

An apparatus and method for encoding and decoding additional information into a digital information in an integral manner. More particularly, the invention relates to a method and device for data protection.

14 Claims, 1 Drawing Sheet


START
ENCODE DIGITAL INFORMATION
ID Portion of Format Information to be Encoded
110
Generate Encoded Format Information
120
Generate Encoded Digital Information
130
DECODE DIGITAL INFORMATION
Decode With Predetermined Key
140
Play Digital Information
150
END

Attachment 20 Page 1 of 1

US007664263B2

(12) **United States Patent**
Moskowitz

(10) **Patent No.: US 7,664,263 B2**
(45) **Date of Patent: Feb. 16, 2010**

(54) **METHOD FOR COMBINING TRANSFER FUNCTIONS WITH PREDETERMINED KEY CREATION**

(76) Inventor: **Scott A. Moskowitz**, 16711 Collins Ave., #2505, Miami, FL (US) 33160

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1222 days.

(21) Appl. No.: **10/602,777**

(22) Filed: **Jun. 25, 2003**

(65) **Prior Publication Data**

US 2004/0086119 A1 May 6, 2004

Related U.S. Application Data

(63) Continuation of application No. 09/046,627, filed on Mar. 24, 1998, now Pat. No. 6,598,162.

(51) **Int. Cl.**
G06K 9/48 (2006.01)
G06F 3/14 (2006.01)

(52) **U.S. Cl.** **380/205**; 380/206; 380/210; 380/236; 380/239; 713/176

(58) **Field of Classification Search** 380/205, 380/206, 210, 236, 239; 713/176
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,947,825	A	3/1976	Cassada	
3,984,624	A	10/1976	Waggener	
3,986,624	A	10/1976	Cates, Jr. et al.	
4,038,596	A	7/1977	Lee	
4,200,770	A	4/1980	Hellman et al.	
4,218,582	A	8/1980	Hellman et al.	
4,339,134	A	7/1982	Macheel	
4,390,898	A *	6/1983	Bond et al.	380/214
4,405,829	A	9/1983	Rivest et al.	
4,424,414	A	1/1984	Hellman et al.	
4,528,588	A	7/1985	Lofberg	
4,672,605	A	6/1987	Hustig et al.	
4,748,668	A	5/1988	Shamir et al.	
4,789,928	A	12/1988	Fujisaki	
4,827,508	A	5/1989	Shear	
4,876,617	A	10/1989	Best et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

EP	0372601 A1	6/1990

(Continued)

OTHER PUBLICATIONS

Schneier, Bruce, Applied Cryptography, 2nd Ed., John Wiley & Sons, pp. 9-10, 1996.

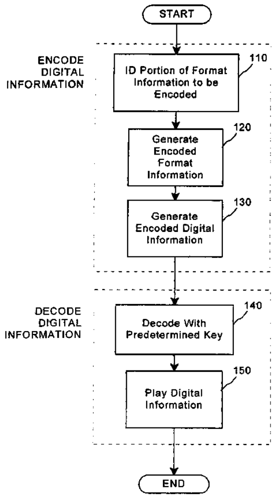
(Continued)

Primary Examiner—Jung Kim
Assistant Examiner—Izunna Okeke

(57) **ABSTRACT**

A method for combining transfer functions with predetermined key creation. In one embodiment, digital information, including a digital sample and format information, is protected by identifying and encoding a portion of the format information. Encoded digital information, including the digital sample and the encoded format information, is generated to protect the original digital information. In another embodiment, a digital signal, including digital samples in a file format having an inherent granularity, is protected by creating a predetermined key. The predetermined key is comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

7 Claims, 1 Drawing Sheet


START
ENCODE DIGITAL INFORMATION
ID Portion of Format Information to be Encoded
110
Generate Encoded Format Information
120
Generate Encoded Digital Information
130
DECODE DIGITAL INFORMATION
Decode With Predetermined Key
140
Play Digital Information
150
END

)

US006598162B1

(12) **United States Patent**
Moskowitz

(10) **Patent No.: US 6,598,162 B1**
(45) **Date of Patent: Jul. 22, 2003**

(54) **METHOD FOR COMBINING TRANSFER FUNCTIONS WITH PREDETERMINED KEY CREATION**

(76) Inventor: **Scott A. Moskowitz**, 16711 Collins Ave. #2505, Miami, FL (US) 33160

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/046,627**

(22) Filed: **Mar. 24, 1998**

Related U.S. Application Data

(63) Continuation-in-part of application No. 08/587,943, filed on Jan. 17, 1996, now Pat. No. 5,745,569.

(51) **Int. Cl.**^{7} **H04L 9/00**; G06F 1/02; G06F 7/58

(52) **U.S. Cl.** **713/176**; 380/46; 708/254

(58) **Field of Search** 713/176; 380/53, 380/54, 46; 708/254

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,200,770	A		4/1980	Hellman et al.	
4,218,582	A		8/1980	Hellman et al.	
4,339,134	A	*	7/1982	Macheel	273/138
4,405,829	A		9/1983	Rivest et al.	
4,424,414	A		1/1984	Hellman et al.	
4,827,508	A		5/1989	Shear	
4,896,275	A	*	1/1990	Jackson	345/668
4,977,594	A		12/1990	Shear	
4,979,210	A		12/1990	Nagata et al.	
5,050,213	A		9/1991	Shear	
5,073,925	A		12/1991	Nagata et al.	
5,369,707	A	*	11/1994	Follendore, III	380/25
5,406,627	A	*	4/1995	Thompson et al.	380/20
5,410,598	A		4/1995	Shear	
5,469,536	A	*	11/1995	Blank	395/131
5,497,419	A	*	3/1996	Hill	
5,513,261	A		4/1996	Maher	
5,530,739	A		6/1996	Okada et al.	
5,530,751	A	*	6/1996	Morris	380/4
5,530,759	A	*	6/1996	Braudaway et al.	380/54
5,598,470	A	*	1/1997	Cooper et al.	380/4

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

EP	0649261	4/1995
NL	100523	9/1998
WO	9744736	11/1997
WO	9952271	10/1999
WO	9963443	12/1999

OTHER PUBLICATIONS

U.S. Patent Appl'n Ser. No. 08/587,943, "Method for Stega–Cipher Protection of Computer Code".

U.S. Patent Appl'n Ser. No. 08/775,216, "Steganographic Method and Device".

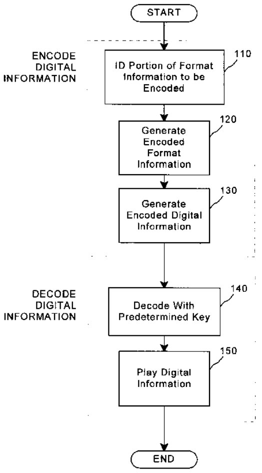
(List continued on next page.)

Primary Examiner—Giberto Barron
Assistant Examiner—Douglas J Meislahn
(74) *Attorney, Agent, or Firm*—Wiley Rein & Fielding LLP

(57) **ABSTRACT**

A method for combining transfer functions with predetermined key creation. In one embodiment, digital information, including a digital sample and format information, is protected by identifying and encoding a portion of the format information. Encoded digital information, including the digital sample and the encoded format information, is generated to protect the original digital information. In another embodiment, a digital signal, including digital samples in a file format having an inherent granularity, is protected by creating a predetermined key. The predetermined key is comprised of a transfer function-based mask set to manipulate data at the inherent granularity of the file format of the underlying digitized samples.

1 Claim, 1 Drawing Sheet


START
ENCODE DIGITAL INFORMATION
ID Portion of Format Information to be Encoded
110
Generate Encoded Format Information
120
Generate Encoded Digital Information
130
DECODE DIGITAL INFORMATION
Decode With Predetermined Key
140
Play Digital Information
150
END

Agreement

Scott Moskowitz agrees to disclose certain information concerning pending patent ("Digital Information Commodities Exchange," filing #083-593, June 30, 1993) owned by Scott Moskowitz.

Marc Cooperman, upon receiving information from whatever source regarding said pending patent, agrees not to disclose or cause to be disclosed any information regarding said pending patent or affect said patent.

Nor shall Marc Cooperman permit any of his/her employees, associates, family members or others to discloseany information concerning said pending patent or cause to be disclosed pending patent in any manner.

Marc Cooperman
Marc Coop
Signature
11/20/93
Date

Scott Moskowitz
Signature
11/12/93
Date

Attachment 23 Page 1 of 2

CERTIFICATE OF REGISTRATION

UNITED STATES COPYRIGHT OFFICE ✱ THE LIBRARY OF CONGRESS ✱

OFFICIAL SEAL

This Certificate issued under the seal of the Copyright Office in accordance with title 17, United States Code, attests that registration has been made for the work identified below. The information on this certificate has been made a part of the Copyright Office records.

Marybeth Peters

REGISTER OF COPYRIGHTS
United States of America

SHORT FORM TX
or a Nondramatic Literary Work
NITED STATES COPYRIGHT OFFICE

TXu 892-516

ffective Date of Registration
Feb 08 1999

Application Received
FEB. 08. 1999

Examined By

Deposit Received
One FEB. 08. 1999 Two

Correspondence □

Fee Received

Amended by C.O. Authority of Scott
skowitz in telephone call on June
, 1999.

TYPE OR PRINT IN BLACK INK. DO NOT WRITE ABOVE THIS LINE.

Title of This Work: **1** Giovanni Master

Alternative title or title of larger work in which this work was published. (audio digital watermark source code)

Name and Address of Author and Owner of the Copyright: **2** Scott Moskowitz
16711 Collins Avenue #2505
Miami Florida 33160

Nationality or domicile. Phone, fax, and email: Phone (305) 956 9041 Fax ( )
Email scott@bluespike.com

Year of Creation: **3** 1998, 1999

***If work has been published,* Date and Nation of Publication:** **4** a. Date Month Day Year (Month, day, and year all required)
b. Nation

Type of Authorship in This Work: **5**
Check all that this author created.

☒ Text (includes fiction, nonfiction, poetry, computer programs, etc.)
□ Illustrations
□ Photographs
* ☒ ~~Compilation of terms or data~~

Signature: **6** *I certify that the statements made by me in this application are correct to the best of my knowledge.* *Check one
Registration cannot be completed without a signature.

☒ Author □ Authorized agent

X Scott Moskowitz

ORIGINAL

Name and Address of Person to Contact for Rights and Permissions: **7** ☒ Check here if same as #2 above
Phone, fax, and email: Phone ( ) Fax ( )
Email

8 Certificate will be mailed in window envelope to this address:

Name ▼ Scott Moskowitz
Number/Street/Apt ▼ 16711 Collins Avenue #2505
City/State/ZIP ▼ Miami Florida 33160

Complete this space only if you currently hold a Deposit Account in the Copyright Office.

9 Deposit Account #______
Name______

DO NOT WRITE HERE Page 1 of ____ pages

*17 U.S.C. § 506(e): Any person who knowingly makes a false representation of a material fact in the application for copyright registration provided for by section 409, or in any written statement filed in connection with the application, shall be fined not more than $2,500.
September 1997—100,000

PRINTED ON RECYCLED PAPER

U.S. COPYRIGHT OFFICE WWW MARCH 1998

Attachment 23 Page 2 of 2

415-443-6811 ATTN: TIM STEVENS
HAPPY B'DAY
FR: S. MOSKOWITZ

November 18, 1996

Dear Scott,

In the past several days I have been trying to communicate with you. Unfortunately, each time seems to end in a shouting match. So this is the only way I know of getting through to you in a clear manner, without being told to shut up long before I get to the point. I want you to do your best to read this through and consider what I am saying with a clear head, and do not jump to conclusions that I am an asshole. Please read it all. It will be long and wandering, but I want you to know everything in my head, because that is the only way to clear this up, however that happens. I know you have a meeting with Joi today, so this can wait until afterward.

I want you to know right here, at the top, before I raise any other issues, that I want this company to move forward, and I want to be part of it. I did not spend two years of my life with the intention of wasting it, and I personally don't want to let you down, because you trusted me. I am not stupid. I can see the threshold of what we are standing on. I am not selfish. I know other people are involved, but that doesn't mean I should fuck myself. I want you to know I do think about other people at the start of this, because the rest of it is simply what is in MY head, and so therefore focuses on me and you.

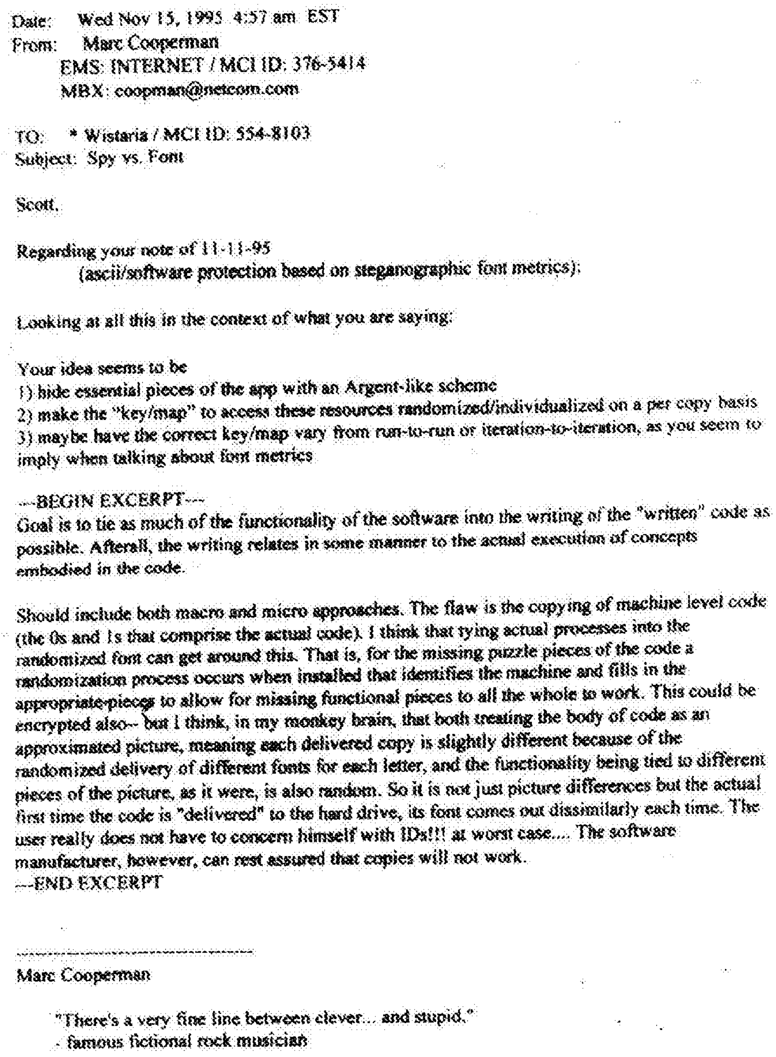
Perhaps my choice of wordage regarding the equity financing clause and investments was a mistake. It has obviously upset you, and you must realize that this is about the last thing I want to do. Think about it. Imagine I am the most selfish person in the world. What could I possibly gain by doing that? I don't want to be in a pissing contest with you. Hindsight is 20/20. Do you think I want to fuck everything up when we are on the verge of success?

I used the term "wordage" because it is precisely that. I took no action whatsoever, and I want that to be clear. Despite what you may think, I have said only good things about you and this company to any investors I have spoken to. I think I have tried on certain occasions to tell you personally how good a management job you are doing, because I had questioned it in the past. I did feel there were certain risks that they had to know about, or I am not meeting an obligation to them. There were certain people on my side of the financing who simply didn't belong there, and that was my mistake, and they pulled out because they were not close enough to me to simply take my word with no other limitations. I said what I said to Tim because he needed to know at the time, what I was thinking, and I did not want to conceal anything from him. Do you think I wanted to raise the issue with him? Of course not, but I felt it would be assholish and unprofessional not to say it at that time.

Now, as a result of what I SAID to Tim directly and straightforwardly, with the intent of reaching you (without waking you up) as soon as possible, I believe you think I am playing games with you and/or trying to hold you hostage? My intent in communicating to you my feelings on the F&H financing and the equity financing clause was to make clear to you that I wanted the issue I raised several weeks ago with you taken care of, one way or the other, before we move forward. I know there were delays, and I did not feel it was such a big deal, considering we had both agreed it was not necessary, which you now tell me has changed, in the interest of buying time, to simply reduce my side of the investment to make it a non-issue for the moment. I am not impatient. I know you are busy, but if you will listen to my reasons below, you will see that I felt it could not remain silent longer. My hope was that we could settle what was on the table between us so it would not require me to do any such thing. We have had numerous arguments about the same issue in the past, and each time you put me off with some variation of "there is no point in talking about this now". So, based on past experience, I felt I had to make it very clear, using the only means

Attachment 24 Page 1 of 1

[redacted]



Date: Wed Nov 15, 1995 4:57 am EST
From: Marc Cooperman
EMS: INTERNET / MCI ID: 376-5414
MBX: coopman@netcom.com

TO: * Wistaria / MCI ID: 554-8103
Subject: Spy vs. Font

Scott,

Regarding your note of 11-11-95
(ascii/software protection based on steganographic font metrics):

Looking at all this in the context of what you are saying:

Your idea seems to be
1) hide essential pieces of the app with an Argent-like scheme
2) make the "key/map" to access these resources randomized/individualized on a per copy basis
3) maybe have the correct key/map vary from run-to-run or iteration-to-iteration, as you seem to imply when talking about font metrics

---BEGIN EXCERPT---
Goal is to tie as much of the functionality of the software into the writing of the "written" code as possible. Afterall, the writing relates in some manner to the actual execution of concepts embodied in the code.

Should include both macro and micro approaches. The flaw is the copying of machine level code (the 0s and 1s that comprise the actual code). I think that tying actual processes into the randomized font can get around this. That is, for the missing puzzle pieces of the code a randomization process occurs when installed that identifies the machine and fills in the appropriate-pieces to allow for missing functional pieces to all the whole to work. This could be encrypted also-- but I think, in my monkey brain, that both treating the body of code as an approximated picture, meaning each delivered copy is slightly different because of the randomized delivery of different fonts for each letter, and the functionality being tied to different pieces of the picture, as it were, is also random. So it is not just picture differences but the actual first time the code is "delivered" to the hard drive, its font comes out dissimilarly each time. The user really does not have to concern himself with IDs!!! at worst case.... The software manufacturer, however, can rest assured that copies will not work.
---END EXCERPT

Marc Cooperman

"There's a very fine line between clever... and stupid."
- famous fictional rock musician

Attachment 25 Page 1 of 1

[redacted]

[redacted]

The code and data resources which comprise the application are compiled and assembled into a collection which is a pre-cursor of an executable application.

At this point, a utility application is used for final assembly of the executable application. The utility will choose one or several essential code resources, and encode them into one or several data resources using the stega-cipher process. The end result will be that these essential code resources are not stored in their own partition, but rather stored as encoded information in data resources.

For the encoding, a key is needed. The purpose of this scheme is to make a licensed copy of an application distinguishable from any other. It is not necessary to distinguish every instance of an application, merely every instance of a license. A licensed user may wish to install multiple copies of an application, legally.

One method then, is to choose the key so that it corresponds, is equal to, or is a function of a license code.

The assembly utility can be supplied with a key generated from a license code generated for the license in question. Given the key, it encodes one or several essential resources into one or several data resources. Exactly which code resources are encoded into which data resources may be determined at random.

Note that the application contains a code resource which performs the function of decoding an encoded code resource from a data resource.

The application must also contain a data resource which specifies in which data resource a particular code resource is encoded.

This data resource is created and added at assembly time by the assembly utility.

The application can then operate as follows. When it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. It stores this information in a personalization data resource.

Once it has the license code, it can then generate the proper decoding key to access the essential code resources.

Note that the application can be copied in an uninhibited manner, but MUST contain the license code issued to the licensed owner, in order to access its essential code resources.

The object of the invention is thus accomplished.

2) Randomly reorganize program memory structure to prevent attempts at memory capture or object code analysis.

The object of this invention is to make it extremely difficult to perform memory capture based analysis of an executable computer program.

Once the code resources of a program are loaded into memory, they typically remain in a fixed position, unless the computer operating system finds it necessary to rearrange certain portions of memory during "system time". Typically, this is done in low memory systems, to maintain optimal memory allocation. If a computer program contains countermeasures against unlicensed copying, a skilled technician can often take a snapshot of the code in memory, analyze it, determine which instructions comprise the countermeasures, and disable them in the stored application file, by means of a "patch".

Under this scheme, the application contains a special code resource which knows about all the other code resources in memory. During execution time, the special code resource, called a "memory scheduler", can be called periodically, or at random intervals, at which time it intentionally shuffles the other code resources randomly in memory, so that someone trying to analyze snapshots of memory at various intervals

[redacted]

Attachment 26 Page 1 of 3

Security schemes for executable computer programs

Background & Info:

An executable computer program is variously referred to as an application, from the point of a user, or executable object code from the point of the engineer, a collection of smaller, atomic (or indivisible) chunks of object code typically comprise a complete executable or application. These indivisible portions of object code correspond with the programmers' function or procedure implementations in higher level languages, such as C or Pascal. In creating an application, a programmer writes "code" in a higher level language, which is then compiled down into "machine language", or, the executable object code, which can actually be run by the computer. Each function, or procedure, written in the programming language, represents a self-contained portion of the larger program, which implements, typically, a very small piece of its functionality. The order in which the programmer types the code for the various functions or procedures, and the distribution of and arrangement of these implementations in various files which hold them is unimportant. Within a function or procedure, however, the order of individual language constructs, which correspond to particular machine instructions is important, and so functions or procedures are considered indivisible for purposes of this discussion. That is, once a function or procedure is compiled, the order of the machine instructions which comprise the executable object code of the function is important and their order in the computer memory is of vital importance. Note that many compilers perform "optimizations" within functions or procedures, which determine, on a limited scale, if there is a better arrangement for executable instructions which is more efficient than that constructed by the programmer, but does not change the result of the function or procedure. Once these optimizations are performed, however, making random changes to the order of instructions is very likely to "break" the function. When a program is compiled, then, it consists of a collection of these sub-objects, whose exact order or arrangement in memory is not important, so long as any sub-object which uses another sub-object knows where in memory it can be found.

The memory address of the first instruction in one of these sub-objects is called the "entry point" of the function or procedure. The rest of the instructions comprising that sub-object immediately follow from the entry point.

These sub-objects can be packaged into what are referred to in certain systems as "code resources", which may be stored separately from the application, or shared with other applications, although not necessarily.

Within an application there are also data objects, which consist of some data to be operated on by the executable code. These data objects are not executable. That is, they do not consist of executable instructions. The data objects can be referred to in certain systems as "resources".

1) hiding necessary parts/resources in digitized sample resources using stega-cipher process

The basic idea in this scheme is that there are a certain sub-set of executable code resources which comprise an application which are essential to the proper function of the application. In general, any code resource can be considered "essential" in that if the program proceeds to a point where it must "call" the code resource, and the code resource is not present in memory, or cannot be loaded, then the program fails. However, we use a definition of "essential" which is more narrow. A program may be written to work around unavailable resources. Particularly programs which incorporate an optional "plug-in architecture", where several code resources may be made optionally available at run-time. We are also concerned with concentrated efforts by technically skilled people who can analyze executable object code and "patch" it to ignore or bypass certain code resources. For our purposes, "essential" means that the function which distinguishes this application from any other application depends upon the presence and use of the code resource in question. The best candidates for this type of code resources are NOT optional, or plug-in types.

Given that there are one or more of these essential resources, we first need the presence of certain data resources of a type which are amenable to the stega-cipher process. Data which consists of image or audio samples is particularly useful.

Attachment 26 Page 2 of 3

[redacted]

cannot be sure if they are looking at the same code or organization from one "break" to the next. This adds significant complexity to their job.

It is also necessary, to complete the effectiveness of the scheme to provide a second special code resource, which knows where the memory scheduler is in memory. This is a "scheduler envelope". It calls the scheduler, and, when the scheduler is finished, it randomly moves the scheduler, since the scheduler cannot move itself, a very hairy operation.

An alternative method is to increase the functionality of the scheduler so that it can move itself. In order to do this, the scheduler would have to first copy itself to a new location, and then specifically modify the program counter and stack frame, so that it could then jump into the new copy of the scheduler, but return to the correct calling frame.

The method described above accomplishes the purpose of the invention to make it hard to analyze captured memory containing application executable code.

12-22-95
Scott Moskowitz

[redacted]

Attachment 26 Page 3 of 3

[redacted]

METHOD FOR STEGA-CIPHER PROTECTION OF COMPUTER CODE

FIELD OF INVENTION

With the advent of computer networks and digital multimedia, protection of intellectual property has become a prime concern for creators and publishers of digitized copies of copyrightable works, such as musical recordings, movies, video games, and computer software. One method of protecting copyrights in the digital domain is to use "digital watermarks." Digital watermarks can be used to mark each individual copy of a digitized work with information identifying the title, copyright holder, and even the licensed owner of a particular copy. When marked with licensing and ownership information, responsibility is created for individual copies where before there was none. Computer application programs can be watermarked by watermarking digital content resources contained within the program, such as images or audio. Digital watermarks can be encoded with random or pseudo random keys, which act as secret maps for locating the watermarks. These keys make it impossible for a party without the key to find the watermark - in addition, the encoding method can be enhanced to force a party to cause damage to a watermarked data stream when trying to erase a random-key watermark. For more information on digital watermarks see
(state specific references, not application serial numbers).
"Steganographic Method and Device" - The DICE Company.
 patent application
"Technology: Digital Commerce", Denise Caruso, New York Times,
 August 7, 1995
"Copyrighting in the Information Age", Harley Ungar,
 ONLINE MARKETPLACE, September 1995, Jupiter Communications

For more information on other methods for hiding information signals in content signals, see
 U.S. Patent No. 5,319,735 - Preuss et al.
 U.S. Patent No. 5,379,345 - Greenberg

It is desirable to use a "stega-cipher" or watermarking process to hide the necessary parts or resources of the executable object code in the digitized sample resources. It is also desirable to further modify the underlying structure of an executable computer application such that it is more resistant to attempts at patching and analysis by memory capture. Being that a computer application seeks to provide a user with certain utilities or tools, that is, users interact with a computer or similar device to accomplish various tasks and applications provide the relevant interface, a level of authentication can also be introduced into software, or "digital products," that include digital content, such as audio, video, pictures or multimedia, with digital watermarks. Security is maximized because erasing this code watermark without a key results in the destruction of one or more essential parts of the underlying application, rendering the "program" useless to the unintended user who lacks the appropriate key. Further, if the key is linked to a license code by means of a mathematical function, a mechanism for identifying the licensed owner of an application is created.

It is also desirable to randomly reorganize program memory structure intermittently during program run time, in order to prevent attempts at memory capture or object code analysis aimed at eliminating licensing or ownership information, or otherwise modifying, in an unintended manner, the functioning of the application. In this way, attempts to capture memory to determine underlying functionality or provide a "patch" to facilitate unauthorized use of the "application," or computer program, can be made difficult or impossible without destroying the functionality and thus usefulness of a copyrightable computer program.

It is thus the goal of the present invention to provide a higher level of copyright security to object code on par with methods described in digital watermarking systems for digitized media content such as pictures, audio, video and multimedia content in its multifarious forms, as described in previous disclosures. "Steganographic Method and Device" and "Human Assisted Random Key Generation and Application for Digital Watermark System." It is a further goal of the present invention to establish methods of copyright protection that can be combined with such schemes as software metering, network

Attachment 27 Page 1 of 8

[redacted]

[redacted]

distribution of code and specialized protection of software that is designed to work over a network, such as that proposed by Sun Microsystems in their HotJava browser and Java programming language, and manipulation of application code in proposed distribution of documents that can be exchanged with resources or the look and feel of the document being preserved over a network, such systems are currently being offered by companies including Adobe, with their Acrobat software. The latter goal being accomplished primarily by means of the watermarking of font, or typeface, resources included in applications or documents, which determine how a bitmap representation of the document is ultimately drawn on a presentation device.

SUMMARY OF THE INVENTION

The present invention includes an application of the technology of "digital watermarks." As described in previous disclosures, "Steganographic Method and Device" and "Human Assisted Random Key Generation and Application for Digital Watermark System," watermarks are particularly suitable to the identification, metering, distributing and authenticating digitized content such as pictures, audio, video and derivatives thereof under the description of "multimedia content." With methods described for combining both cryptographic methods, and steganography, or hiding something in plain view. Discussions of these technologies can be found in Applied Cryptography by Bruce Schneier and The Code Breakers by David Kahn. For more information on prior art public-key cryptosystems see US Pat No 4,200,770 Diffie-Hellman, 4,218,582 Hellman, 4,405,829 RSA, 4,424,414 Hellman Pohlig. Computer code, or machine language instructions, which are not digitized and have zero tolerance for error, must be protected by derivative or alternative methods, such as those disclosed in this invention, which focuses on watermarking with "keys" derived from license codes or other ownership identification information, and using the watermarks encoded with such keys to hide an essential sub set of the application code resources.

It is thus a goal of the present invention, to provide a level of security for executable code on similar grounds as that which can be provided for digitized samples. The prior art includes copy protection systems attempted at many stages in the development of the software industry, these may be various methods by which a software engineer can write the software in a clever manner to determine if it has been copied, and if so to deactivate itself. Also included are undocumented changes to the storage format of the content. Copy protection was generally abandoned by the software industry, since pirates were generally just as clever as the software engineers and figured out ways to modify their software and deactivate the protection. The cost of developing such protection was not justified considering the level of piracy which occurred despite the copy protection. Other methods for protection of computer software include the requirement of entering certain numbers or facts that may be included in a packaged software's manual, when prompted at start-up. These may be overcome if copies of the manual are distributed to unintended users, or by patching the code to bypass these measures. Other methods include requiring a user to contact the software vendor and disclosing "keys" for unlocking software after registration attached to some payment scheme, such as credit card authorization. Further methods include network-based searches of a user's hard drive and comparisons between what is registered to that user and what is actually installed on the user's general computing device. Other proposals, by such parties as Bell Labs, use "kerning" or actual distance in pixels, in the rendering of text documents, rather than a varied number of ASCII. However, this approach can often be defeated graphics processing analogous to sound processing, which randomizes that information. All of these methods require outside determination and verification of the validity of the software license. The present invention differs from the prior art in that it does not attempt to stop copying, but rather, to determine responsibility for a copy by ensuring that licensing information must be preserved in descendant copies from an original. Without the correct license information, the copy cannot function.

An improvement over the art is disclosed in the present invention, in that the software itself is a set of commands, compiled by software engineer, which can be configured in such a manner as to tie underlying functionality to the license or authorization of the copy in possession by the user. Without such verification, the functions sought out by the user in the form of software cease to properly work. Attempts to tamper or "patch" substitute code resources can be made highly difficult by randomizing the location of said resources in memory on an intermittent basis to resist most attacks at disabling the system.

BRIEF DESCRIPTION OF THE DRAWINGS

Attachment 27 Page 2 of 8

[redacted]

[redacted]

DETAILED DESCRIPTION

An executable computer program is variously referred to as an application, from the point of a user, or executable object code from the point of the engineer. A collection of smaller, atomic (or indivisible) chunks of object code typically comprise the complete executable object code or application which may also require the presence of certain data resources. These indivisible portions of object code correspond with the programmers' function or procedure implementations in higher level languages, such as C or Pascal. In creating an application, a programmer writes "code" in a higher level language, which is then compiled down into "machine language," or, the executable object code, which can actually be run by a computer, general purpose or otherwise. Each function, or procedure, written in the programming language, represents a self-contained portion of the larger program, and implements, typically, a very small piece of its functionality. The order in which the programmer types the code for the various functions or procedures, and the distribution of and arrangement of these implementations in various files which hold them is unimportant. Within a function or procedure, however, the order of individual language constructs, which correspond to particular machine instructions is important, and so functions or procedures are considered indivisible for purposes of this discussion. That is, once a function or procedure is compiled, the order of the machine instructions which comprise the executable object code of the function is important and their order in the computer memory is of vital importance. Note that many "compilers" perform "optimizations" within functions or procedures, which determine, on a limited scale, if there is a better arrangement for executable instructions which is more efficient than that constructed by the programmer, but does not change the result of the function or procedure. Once these optimizations are performed, however, making random changes to the order of instructions is very likely to "break" the function. When a program is compiled, then, it consists of a collection of these sub-objects, whose exact order or arrangement in memory is not important, so long as any sub-object which uses another sub-object knows where in memory it can be found.

The memory address of the first instruction in one of these sub-objects is called the "entry point" of the function or procedure. The rest of the instructions comprising that sub-object immediately follow from the entry point. Some systems may prefix information to the entry point which describes calling and return conventions for the code which follows, an example is the Apple Macintosh Operating System (MacOS). These sub-objects can be packaged into what are referred to in certain systems as "code resources," which may be stored separately from the application, or shared with other applications, although not necessarily. Within an application there are also data objects, which consist of some data to be operated on by the executable code. These data objects are not executable. That is, they do not consist of executable instructions. The data objects can be referred to in certain systems as "resources."

It is a goal, in seeking to purchase or acquire a computer program, by a user that a computer program "function" in a some desired manner. Simply, computer software is overwhelmingly purchased for its underlying functionality. In contrast, persons who copy multimedia content, such as pictures, audio and video, do so for the entertainment or commercial value of the content. The difference between the two types of products is that multimedia content is not generally interactive, but passive, and its commercial value relates more on passive not interactive or utility features, such as that required in packaged software, set-top boxes, cellular phones, VCRs, PDAs, and the like. Simply, interactive digital products which include computer code may be mostly interactive but can also contain content to add to the interactive experience of the user or make the underlying utility of the software more aesthetically pleasing. It is a common concern of both of these creators, both of interactive and passive multimedia products, that "digital products" can be easily and perfectly copied and made into unpaid or unauthorized copies. This concern is especially heightened when the underlying product is copyrighted and intended for commercial use.

The first method described in the present invention involves hiding necessary "parts" or "resources" in digitized sample resources using "digital watermarking" process, such as that described in the "Steganographic Method and Device" patent application. The basic premise for this scheme is that there are a certain sub-set of executable code resources, which comprise an application, that are "essential" to the proper function of the application. In general, any code resource can be considered "essential" in that if the program proceeds to a point where it must "call" the code resource, and the code resource is not present in memory, or cannot be loaded, then the program fails. However, the present invention uses a

Attachment 27 Page 3 of 8

[redacted]

[redacted]

definition of "essential" which is more narrow. This is because, those skilled in the art or those with programming experience, may create a derivative program, not unlike the utility provided by the original program, by writing additional or substituted code to work around unavailable resources. This is particularly true with programs that incorporate an optional "plug-in architecture," where several code resources may be made optionally available at run-time. The present invention is also concerned with concentrated efforts by technically skilled people who can analyze executable object code and "patch" it to ignore or bypass certain code resources. Thus, for the present embodiment's purposes, "essential" means that the function which distinguishes this application from any other application depends upon the presence and use of the code resource in question. The best candidates for this type of code resources are NOT optional, or plug-in types, unless special care is taken to prevent work-a-rounds.

Given that there are one or more of these essential resources, what is needed to realize the present invention is the presence of certain data resources of a type which are amenable to the "stega-cipher" process described in the "Steganographic Method and Device" patent application. Data which consists of image or audio samples is particularly useful. Because this data consists of digital samples, digital watermarks can be introduced into the samples. What is further meant is that certain applications include image and audio samples which are important to the look and feel of the program or are essential to the processing of the application's functionality when used by the user. These computer programs are familiar to users of computers but also less obvious to users of other devices that run applications that are equivalent in some measure of functionality to general purpose computers including, but not limited to, set-top boxes, cellular phones, "smart televisions," PDAs and the like. However, programs still comprise the underlying "operating systems" of these devices and are becoming more complex with increases in functionality.

One method of the present invention is now discussed. When code and data resources are compiled and assembled into a precursor of an executable program the next step is that a utility application is used for final assembly of the executable application. The utility will choose one or several essential code resources, and encode them into one or several data resources using the stega-cipher process. The end result will be that these essential code resources are not stored in their own partition, but rather stored as encoded information in data resources. They are not accessible at run-time without the key. Basically, the essential code resources that provide functionality in the final end-product, an executable application or computer program, are no longer easily and recognizably available for manipulation by those seeking to remove the underlying copyright or license, or its equivalent information, or those with skill to substitute alternative code resources to "force" the application program to run as an unauthorized copy. For the encoding of the essential code resources, a "key" is needed. Such a key is similar to those described in the "Steganographic Method and Device." The purpose of this scheme is to make a licensed copy of an application distinguishable from any other. It is not necessary to distinguish every instance of an application, merely every instance of a license. A licensed user may then wish to install multiple copies of an application, legally or with authorization. This method, then, is to choose the key so that it corresponds, is equal to, or is a function of, a license code, not just a text file, audio clip or identifying piece of information as desired in digital watermarking schemes extant and typically useful to stand-alone, digitally sampled content. The key is necessary to access to underlying code, what the user understands to be the application program.

The assembly utility can be supplied with a key generated from a license code generated for the license in question. Given the key, it encodes one or several essential resources into one or several data resources. Exactly which code resources are encoded into which data resources may be determined in a random or pseudo random manner. Note further that the application contains a code resource which performs the function of decoding an encoded code resource from a data resource. The application must also contain a data resource which specifies in which data resource a particular code resource is encoded. This data resource is created and added at assembly time by the assembly utility. The application can then operate as follows:

1) When it is run for the first time, after installation, it asks the user for personalization information, which includes the license code. This can include a particular computer configuration.
2) It stores this information in a personalization data resource.

[redacted]

[redacted]

3) Once it has the license code, it can then generate the proper decoding key to access the essential code resources.

Note that the application can be copied in an uninhibited manner, but must contain the license code issued to the licensed owner, in order to access its essential code resources. The goal of the invention, copyright protection of computer code and establishment of responsibility for copies, is thus accomplished.

This invention represents a significant improvement over prior art because of the inherent difference in use of purely informational watermarks versus watermarks which contain executable object code. If the executable object code in a watermark is essential to an application which accesses the data which contains the watermark, this creates an all-or-none situation. Either the user must have the extracted watermark, or the application cannot be used, and hence the user cannot gain full access to the presentation of the information in the watermark bearing data. In order to extract a digital watermark, the user must have a key. The key, in turn, is a function of the license information for the copy of the software in question. The key is fixed prior to final assembly of the application files, and so cannot be changed at the option of the user. That, in turn, means the license information in the software copy must remain fixed, so that the correct key is available to the software. The key and the license information are, in fact, interchangeable. One is merely more readable than the other. In the prior art, "Steganographic Method and Device," the possibility of randomization erasure attacks on digital watermarks was discussed. Simply, it is always possible to erase a digital watermark, depending on how much damage you are willing to do to the watermark-bearing content stream. The present invention has the significant advantage that you must have the watermark to be able to use the code it contains. If you erase the watermark you have lost a key piece of the functionality of the application, or even the means to access the data which bears the watermark.

A preferred embodiment would be implemented in an embedded system, with a minimal operating system and memory. No media playing "applets," or smaller sized applications as proposed in new operating environments envisioned by Sun Microsystems and the advent of Sun's Java operating system, would be permanently stored in the system, only the bare necessities to operate the device, download information, decode watermarks and execute the applets contained in them. When an applet is finished executing, it is erased from memory. Such a system would guarantee that content which did not contain readable watermarks could not be used. This is a powerful control mechanism for ensuring that content to be distributed through such a system contains valid watermarks. Thus, in such networks as the Internet or set-top box controlled cable systems, distribution and exchange of content would be made more secure from unauthorized copying to the benefit of copyright holders and other related parties. The system would be enabled to invalidate, by default, any content which has had its watermark(s) erased, since the watermark conveys, in addition to copyright information, the means to fully access, play, record or otherwise manipulate, the content.

A second method for the present invention is to randomly re-organize program memory structure to prevent attempts at memory capture or object code analysis. The object of this method is to make it extremely difficult to perform memory capture-based analysis of an executable computer program. This analysis is the basis for a method of attack to defeat the system envisioned by the present invention.

Once the code resources of a program are loaded into memory, they typically remain in a fixed position, unless the computer operating system finds it necessary to rearrange certain portions of memory during "system time," when the operating system code, not application code, is running. Typically, this is done in low memory systems, to maintain optimal memory utilization. The MacOS for example, uses Handles, which are double-indirect pointers to memory locations, in order to allow the operating system to rearrange memory transparently, underneath a running program. If a computer program contains countermeasures against unlicensed copying, a skilled technician can often take a snapshot of the code in memory, analyze it, determine which instructions comprise the countermeasures, and disable them in the stored application file, by means of a "patch." Other applications for designing code that moves to prevent scanning-tunnelling microscopes, and similar high sensitive hardware for analysis of electronic structure of microchips running code, have been proposed by such parties as Wave Systems. Designs of Wave Systems' microchip are intended for preventing attempts by hackers "photograph" or otherwise determine "burn in" to microchips for attempts at reverse engineering. The present invention seeks to prevent

Attachment 27 Page 5 of 8

[redacted]

[redacted]

attempts at patches that can be introduced to determine the code that comprises the application file. Unlike systems such as Wave Systems', the present invention seeks to move code around in such a manner as to complicate attempts by software engineers to reengineer a means to disable the methods for creating licensed copies on any device that lacks "trusted hardware." Moreover, the present invention concerns itself with any application software that may be used in general computing devices, not chipsets that are used in addition to an underlying computer to perform encryption. Wave Systems approach to security of software if interpreted similarly to the present invention would dictate separate microchip sets for each piece of application software that would be tamperproof-- not consistent with the economics of software and its distribution.

Under the present invention, the application contains a special code resource which knows about all the other code resources in memory. During execution time, this special code resource, called a "memory scheduler," can be called periodically, or at random or pseudo random intervals, at which time it intentionally shuffles the other code resources randomly in memory, so that someone trying to analyze snapshots of memory at various intervals cannot be sure if they are looking at the same code or organization from one "break" to the next. This adds significant complexity to their job. The scheduler also randomly relocates itself when it is finished. In order to do this, the scheduler would have to first copy itself to a new location, and then specifically modify the program counter and stack frame, so that it could then jump into the new copy of the scheduler, but return to the correct calling frame. Finally, the scheduler would need to maintain a list of all memory addresses which contain the address of the scheduler, and change them to reflect its new location. The methods described above accomplishes the purpose of the invention - to make it hard to analyze captured memory containing application executable code in order to create an identifiable computer program or application that is different from other copies and is less susceptible to unauthorized use by those attempting to disable the underlying copyright protection system. Simply, each copy has particular identifying information making that copy different from all other copies.

What is Claimed:

1) The method of associating executable object code with a digital sample stream by means of a digital watermark wherein the digital watermark contains the executable object code and is encoded into the digital sample stream

2) The method of claim 1 where the key to access the digital watermark is a function of a collection of license information pertaining to the software which is accessing the watermark

where license information consists of one or more of the following items

Owning Organization name
Personal Owner name
Owner Address
License code
Software serialization number
Distribution parameters
Appropriate executable general computing device architecture
Pricing
Software Metering details

3) The method of claim 1 further comprised of the step of transmitting the digital sample stream, via a transmission means, from a publisher to a subscriber

where transmission means can be one of

soft sector magnetic disk media
hard sector magnetic disk media
magnetic tape media
CD-ROM disc media
CD-R disc media
Digital Video Disk media
magneto-optical disk media
memory cartridge

Attachment 27 Page 6 of 8

[redacted]

[redacted]

telephone lines
SCSI
Ethernet or Token Ring Network
ISDN
ATM network
TCP/IP network
analog cellular network
digital cellular network
wireless network
digital satellite
cable network
fiber optic network
electric powerline network

4) The method of claim 1 where the object code to be encoded is comprised of series of executable machine instructions which perform the function of at least one of

(ADD APPLET LANGUAGE HERE)
processing a digital sample stream for the purpose of modifying it
playing a digital sample stream

5) The method of claims 3 and 4 further comprised of the steps of:

decoding said digital watermark and extracting object code
loading object code into computer memory for the purpose of execution
executing said object code in order to process said digital sample stream for
the purpose of playback

6) The method of assembling an application to be protected by watermark encoding of essential resources comprised of the steps of

assembling a list of identifiers of essential code resources of an application
where identifiers allow the code resource to be accessed and loaded into memory
providing license information on the licensee who is to receive an individualized
copy of the application
storing license information in a personalization resource which is added to the list
of application data resources
generating a digital watermark key from the license information
using the key as a psuedo-random number string to select a list of suitable digital
sample data resources, the list of essential code resources, and a mapping of which essential code resources are to be watermarked into which data resources
storing the map, which is a list of paired code and data resource identifiers, as a data resource, which is added to the application
adding a digital watermark decoder code resource to the application, to provide a
means for extracting essential code resource from data resources,
according to the map
processing the map list and encoding essential code resources into digital sample
data resources with a digital watermark encoder
removing self-contained copies of the essential code resources which have been
watermarked into data resources
combining all remaining code and data resources into a single application installer

7) The method of intermittently relocating application code resources in computer memory, in order to prevent, discourage, or complicate attempts at memory capture based code analysis

8) The method of claim 7 additionally comprised of the steps of

assembling a list of identifiers of code resources of an application

[redacted]

[redacted]

where identifiers allow the code resource to be accessed and loaded into memory

9) The method of claim 8 additionally comprised of the step of modifying application program structure to make all code resource calls indirectly, through the memory scheduler, which looks up code resources in its list and dispatches calls

10) The method of claim 9 additionally comprised of the step of intermittently rescheduling or shuffling all code resources prior to or following the dispatch of a code resource call through the memory scheduler

11) The method of claim 10 additionally comprised of the step of the memory scheduler copying itself to a new location in memory

12) The method of claim 11 additionally comprised of the step of modifying the stack frame, program counter, and memory registers of the CPU to cause the scheduler to jump to the next instruction comprising the scheduler, in the copy, to erase the previous memory instance of the scheduler, changing all memory references to the scheduler to reflect its new location, and to return from the copy of the scheduler to the frame which called the previous copy of the scheduler

ABSTRACT:

01-03-96
Scott Moskowitz

[redacted]

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,137	05/11/2018	9021602	90014137	6880

31518 7590 03/27/2019

NEIFELD IP LAW, PC
5400 Shawnee Road
Suite 310
ALEXANDRIA, VA 22312-2300

EXAMINER
WOOD, WILLIAM H

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
03/27/2019	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

FISCH SIGLER LLP
5301 WISCONSIN AVENUE, NW
FOURTH FLOOR
WASHINGTON, DC 20015

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/014,137* .

PATENT UNDER REEXAMINATION *9021602* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Notice of Intent to Issue Ex Parte Reexamination Certificate	**Control No.** 90/014,137	**Patent Under Reexamination** 9021602	
	Examiner WILLIAM H WOOD	**Art Unit** 3992	**AIA Status** No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

1. ☑ Prosecution on the merits is (or remains) closed in this *ex parte* reexamination proceeding. This proceeding is subject to reopening at the initiative of the Office or upon petition. *Cf.* 37 CFR 1.313(a). A Certificate will be issued in view of
   (a) ☑ Patent owner's communication(s) filed: 01/23/2019.
   (b) □ Patent owner's failure to file an appropriate timely response to the Office action mailed: _____.
   (c) □ Patent owner's failure to timely file an Appeal Brief (37 CFR 41.31).
   (d) □ The decision on appeal by the □ Board of Patent Appeals and Interferences □ Court dated _____
   (e) □ Other: _____.
2. The Reexamination Certificate will indicate the following:
   (a) Change in the Specification: □ Yes ☑ No
   (b) Change in the Drawing(s): □ Yes ☑ No
   (c) Status of the Claim(s):
      (1) Patent claim(s) confirmed: 1-5,8,10 and 12.
      (2) Patent claim(s) amended (including dependent on amended claim(s)): _____
      (3) Patent claim(s) canceled: _____.
      (4) Newly presented claim(s) patentable: _____.
      (5) Newly presented canceled claims: _____.
      (6) Patent claim(s) □ previously □ currently disclaimed: _____
      (7) Patent claim(s) not subject to reexamination: 6-7,9,11 and 13-19.
3. □ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.
4. ☑ Note the attached statement of reasons for patentability and/or confirmation. Any comments considered necessary by patent owner regarding reasons for patentability and/or confirmation must be submitted promptly to avoid processing delays. Such submission(s) should be labeled: "Comments On Statement of Reasons for Patentability and/or Confirmation."
5. □ Note attached NOTICE OF REFERENCES CITED (PTO-892).
6. □ Note attached LIST OF REFERENCES CITED (PTO/SB/08 or PTO/SB/08 substitute).
7. □ The drawing correction request filed on _____ is: □approved □disapproved.
8. □ Acknowledgment is made of the priority claim under 35 U.S.C. § 119(a)-(d) or (f).
   a) □ All b) □ Some* c) □None of the certified copies have
   □been received.
   □not been received.
   □been filed in Application No. _____.
   □been filed in reexamination Control No. _____.
   □been received by the International Bureau in PCT Application No. _____.

   * Certified copies not received: _____.
9. □ Note attached Examiner's Amendment.
10. □ Note attached Interview Summary (PTO-474).
11. □ Other: _____.

All correspondence relating to this reexamination proceeding should be directed to the **Central Reexamination Unit** at the mail, FAX, or hand-carry addresses given at the end of this Office action.

/William H. Wood/
Primary Examiner, Art Unit 3992

cc: Requester (if third party requester)

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

STATEMENT OF REASONS FOR PATENTABILITY AND/OR CONFIRMATION

Claim Status

Claims subject to reexamination: 1-5, 8, 10 and 12.

Claims not subject to reexamination: 6-7, 9, 11 and 13-19.

Claims patentable/confirmed: 1-5, 8, 10 and 12.

Statement of Reasons for Patentability and/or Confirmation

The following is an examiner's statement of reasons for patentability and/or confirmation of the claims found patentable in this reexamination proceeding: the cited prior art, including *Cooperman*, *Hicks, Rhoads,* and *Moskowitz et al.*, were previously applied, but are now not eligible as prior art under 102(a), (e), or (g). Patent Owner has established invention by both Scott Moskowitz and Marc Cooperman (see granted petition of 12/12/2018). Further, the declaration under 37 CFR 1.131 (01/23/2019) has been considered and establishes invention prior to *Hicks* and *Rhoades*.

Any comments considered necessary by PATENT OWNER regarding the above statement must be submitted promptly to avoid processing delays. Such submission by the patent owner should be labeled: "Comments on Statement of Reasons for Patentability and/or Confirmation" and will be placed in the reexamination file.

Correspondence Information

All correspondence relating to this *ex parte* reexamination proceeding should be directed:

By Mail to:
Mail Stop *Ex Parte* Reexam
Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

By FAX to:
(571) 273-9900
Central Reexamination Unit

By hand:
Customer Service Window
Attn: Central Reexamination Unit
Randolph Building
401 Dulany Street
Alexandria, VA 22314

By EFS-Web:
Registered users of EFS-Web may alternatively submit correspondence via electronic filing system EFS-Web.

Any inquiry concerning this communication or earlier communications from the Reexamination Legal Advisor or Examiner, or as to the status of this proceeding should be directed to the Central Reexamination Unit at telephone number (571)272-7705.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR systems, see http://pair-direct.uspto.gov. For questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/William H. Wood/
Reexamination Specialist, Art Unit 3992

Conferee:

/RSD/

/ALEXANDER J KOSOWSKI/
Supervisory Patent Examiner, Art Unit 3992

US009021602C1

(12) EX PARTE REEXAMINATION CERTIFICATE (11507th)

United States Patent

Moskowitz

(10) **Number: US 9,021,602 C1**

(45) **Certificate Issued: Apr. 23, 2019**

(54) **DATA PROTECTION METHOD AND DEVICE**

(71) Applicant: **Scott A. Moskowitz**, Sunny Isles Beach, FL (US)

(72) Inventor: **Scott A. Moskowitz**, Sunny Isles Beach, FL (US)

(73) Assignee: **WISTARIA TRADING LTD**, Hamilton (BM)

Reexamination Request:

No. 90/014,137, May 11, 2018

Reexamination Certificate for:

Patent No.:	**9,021,602**
Issued:	**Apr. 28, 2015**
Appl. No.:	**13/794,584**
Filed:	**Mar. 11, 2013**

Certificate of Correction issued Sep. 29, 2015

Related U.S. Application Data

(60) Continuation of application No. 13/556,420, filed on Jul. 24, 2012, now Pat. No. 8,930,719, which is a continuation of application No. 11/895,388, filed on Aug. 24, 2007, now Pat. No. 9,104,842, which is a division of application No. 10/602,777, filed on Jun. 25, 2003, now Pat. No. 7,664,263, which is a continuation of application No. 09/046,627, filed on Mar. 24, 1998, now Pat. No. 6,598,162.

(51) **Int. Cl.**

G06F 21/00	(2013.01)
G06F 21/10	(2013.01)
H04L 9/32	(2006.01)
G06F 21/12	(2013.01)
G06F 21/16	(2013.01)
G06F 21/62	(2013.01)
G06F 21/33	(2013.01)
G06F 21/60	(2013.01)
G06T 1/00	(2006.01)
H04L 9/06	(2006.01)

(52) **U.S. Cl.**

CPC ***G06F 21/10*** (2013.01); ***G06F 21/125*** (2013.01); ***G06F 21/16*** (2013.01); ***G06F 21/335*** (2013.01); ***G06F 21/602*** (2013.01); ***G06F 21/6209*** (2013.01); ***G06T 1/0021*** (2013.01); ***H04L 9/065*** (2013.01); ***H04L 9/3236*** (2013.01); ***H04L 9/3247*** (2013.01); *G06F 2211/007* (2013.01); *G06F 2221/0737* (2013.01); *G06F 2221/2107* (2013.01); *G06T 2201/0064* (2013.01); *G06T 2201/0083* (2013.01); *H04L 2209/605* (2013.01); *H04L 2209/608* (2013.01)

(58) **Field of Classification Search**

None

See application file for complete search history.

(56) **References Cited**

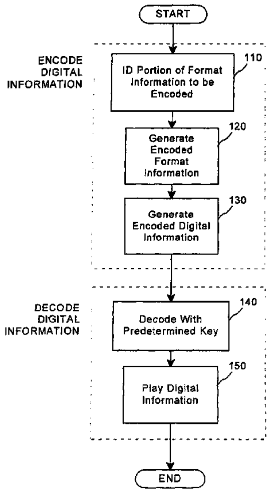
To view the complete listing of prior art documents cited during the proceeding for Reexamination Control Number 90/014,137, please refer to the USPTO's public Patent Application Information Retrieval (PAIR) system under the Display References tab.

Primary Examiner — William H. Wood

(57) **ABSTRACT**

An apparatus and method for encoding and decoding additional information into a digital information in an integral manner. More particularly, the invention relates to a method and device for data protection.

Attention is directed to the decision of 5:18cv3392; 1:18cv1427; 6:18cv242; 1:18cv1406; 1:18cv1512 relating to this patent. This reexamination may not have resolved all questions raised by this decision. See 37 CFR 1.552(c) for *ex parte* reexamination and 37 CFR 1.906(c) for *inter partes* reexamination.


START
ENCODE DIGITAL INFORMATION
ID Portion of Format Information to be Encoded
110
Generate Encoded Format Information
120
Generate Encoded Digital Information
130
DECODE DIGITAL INFORMATION
Decode With Predetermined Key
140
Play Digital Information
150
END

EX PARTE REEXAMINATION CERTIFICATE

NO AMENDMENTS HAVE BEEN MADE TO THE PATENT

AS A RESULT OF REEXAMINATION, IT HAS BEEN DETERMINED THAT:

The patentability of claims **1**-**5**, **8**, **10** and **12** is confirmed.

Claims **6**-**7**, **9**, **11** and **13**-**19** were not reexamined.

* * * * *