US 20150016663A1

(19) **United States**

(12) **Patent Application Publication**

Tehranchi et al.

(10) **Pub. No.: US 2015/0016663 A1**

(43) **Pub. Date: Jan. 15, 2015**

(54) **WATERMARKING IN AN ENCRYPTED DOMAIN**

(71) Applicant: **Verance Corporation**, San Diego, CA (US)

(72) Inventors: **Babak Tehranchi**, San Diego, CA (US); **Rade Petrovic**, San Diego, CA (US)

(21) Appl. No.: **14/329,757**

(22) Filed: **Jul. 11, 2014**

Related U.S. Application Data

(63) Continuation of application No. 11/482,519, filed on Jul. 7, 2006, now Pat. No. 8,781,967.

(60) Provisional application No. 60/697,515, filed on Jul. 7, 2005.

Publication Classification

(51) **Int. Cl.**

G06T 1/00	(2006.01)
H04N 21/8358	(2006.01)
H04N 21/4405	(2006.01)
H04N 1/32	(2006.01)
H04N 21/2347	(2006.01)

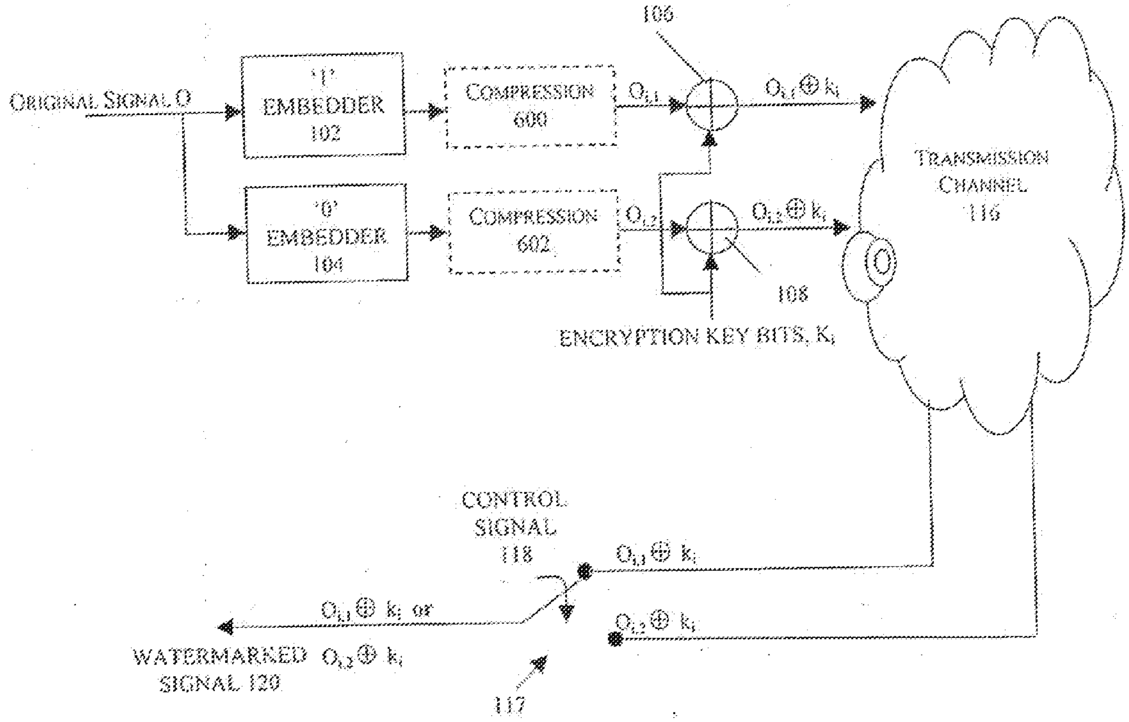
(52) **U.S. Cl.**

CPC ***G06T 1/0035*** (2013.01); ***H04N 1/32272*** (2013.01); ***H04N 21/2347*** (2013.01); ***H04N 21/4405*** (2013.01); ***H04N 21/8358*** (2013.01); *G06T 2201/0053* (2013.01)

USPC **382/100**

(57) **ABSTRACT**

Methods, apparatus and systems for embedding auxiliary information in encrypted host signals are provided. The present invention enables secure application of digital watermarks at any point in the transmission and/or distribution of digital content by enabling the insertion of a plurality of digital watermarks, without the knowledge of the encryption/decryption keys, into a digital host content that has been encrypted with an encryption key. The embedded watermarks persist throughout the content subsequent to the decryption of the content. The disclosed techniques are applicable to content that has been encrypted using a variety of different encryption techniques and algorithms, including stream ciphers, block ciphers, symmetric and asymmetric encryption algorithms. These methods are further adapted to enable the insertion of watermarks into a content that is compressed prior to encryption.



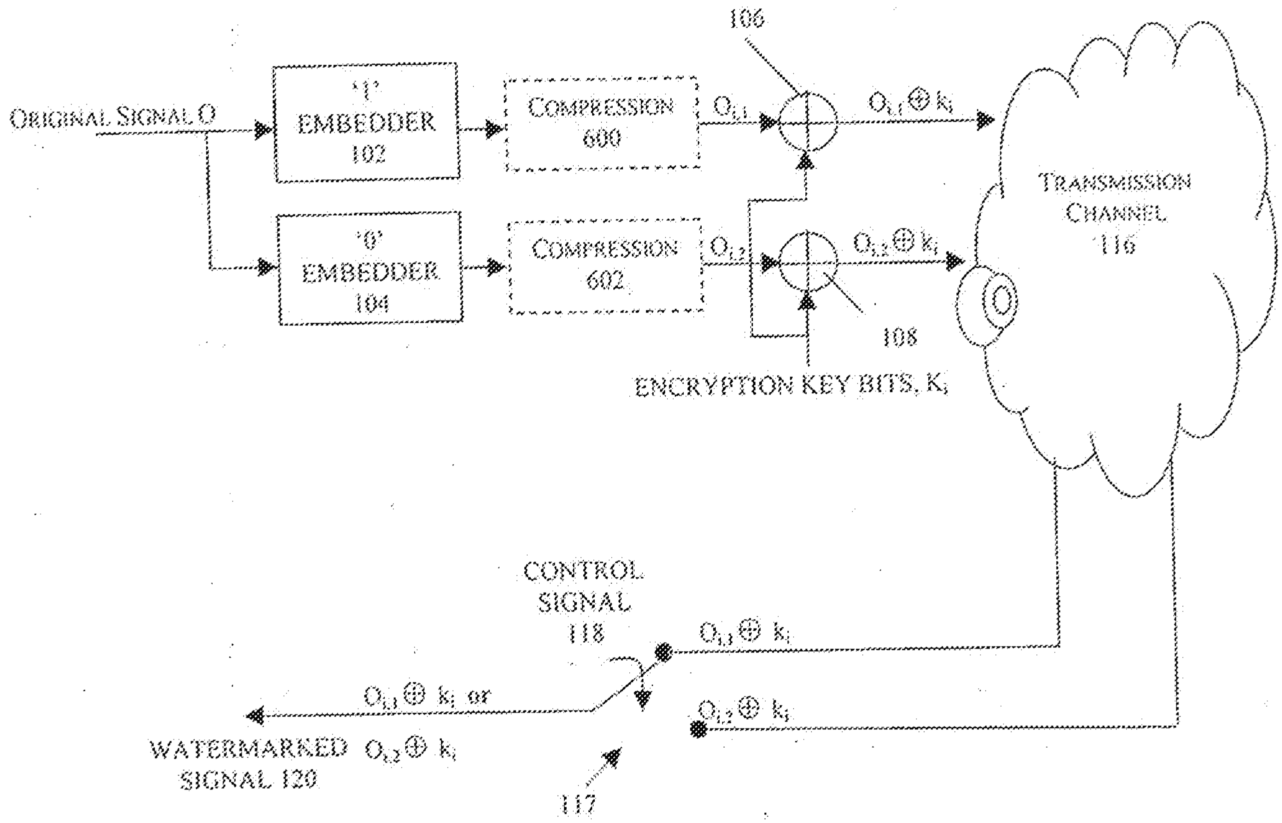

ORIGINAL SIGNAL O
'1' EMBEDDER 102
'0' EMBEDDER 104
COMPRESSION 600
COMPRESSION 602
$O_{i,1}$
$O_{i,2}$
106
108
$O_{i,1} \oplus k_i$
$O_{i,2} \oplus k_i$
ENCRYPTION KEY BITS, K_i
TRANSMISSION CHANNEL 116
CONTROL SIGNAL 118
$O_{i,1} \oplus k_i$
$O_{i,2} \oplus k_i$
$O_{i,1} \oplus k_i$ or
WATERMARKED SIGNAL 120 $O_{i,2} \oplus k_i$
117

FIG. 1

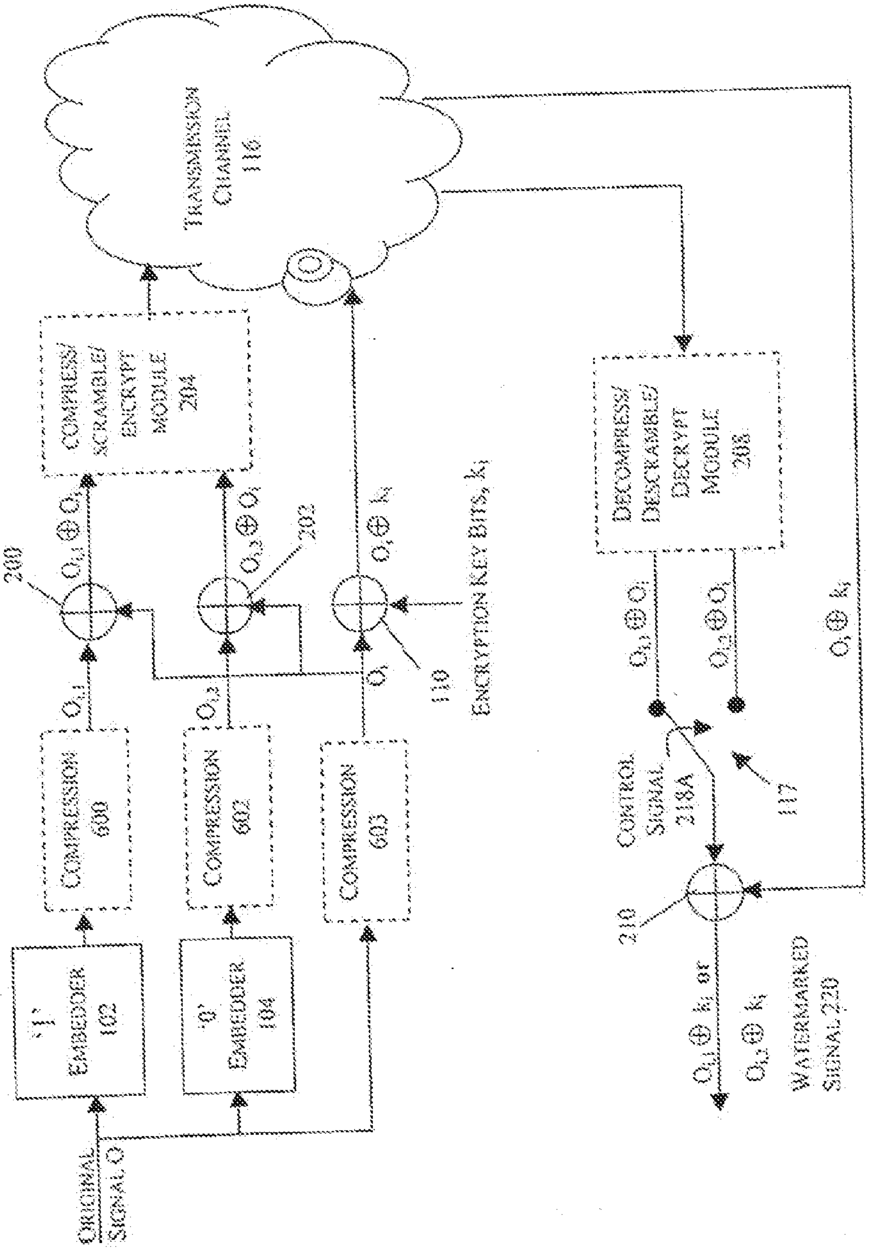

ORIGINAL SIGNAL O
'1' EMBEDDER 102
'0' EMBEDDER 104
COMPRESSION 600
COMPRESSION 602
COMPRESSION 603
$O_{i,1}$
$O_{i,0}$
O_i
200
202
110
$O_{i,1} \oplus O_i$
$O_{i,0} \oplus O_i$
$O_i \oplus k_i$
ENCRYPTION KEY BITS, k_i
COMPRESS/ SCRAMBLE/ ENCRYPT MODULE 204
TRANSMISSION CHANNEL 116
DECOMPRESS/ DESCRAMBLE/ DECRYPT MODULE 208
$O_{i,1} \oplus O_i$
$O_{i,0} \oplus O_i$
$O_i \oplus k_i$
CONTROL SIGNAL 218A
117
210
$O_{i,1} \oplus k_i$ or $O_{i,0} \oplus k_i$
WATERMARKED SIGNAL 220

Fig. 2A

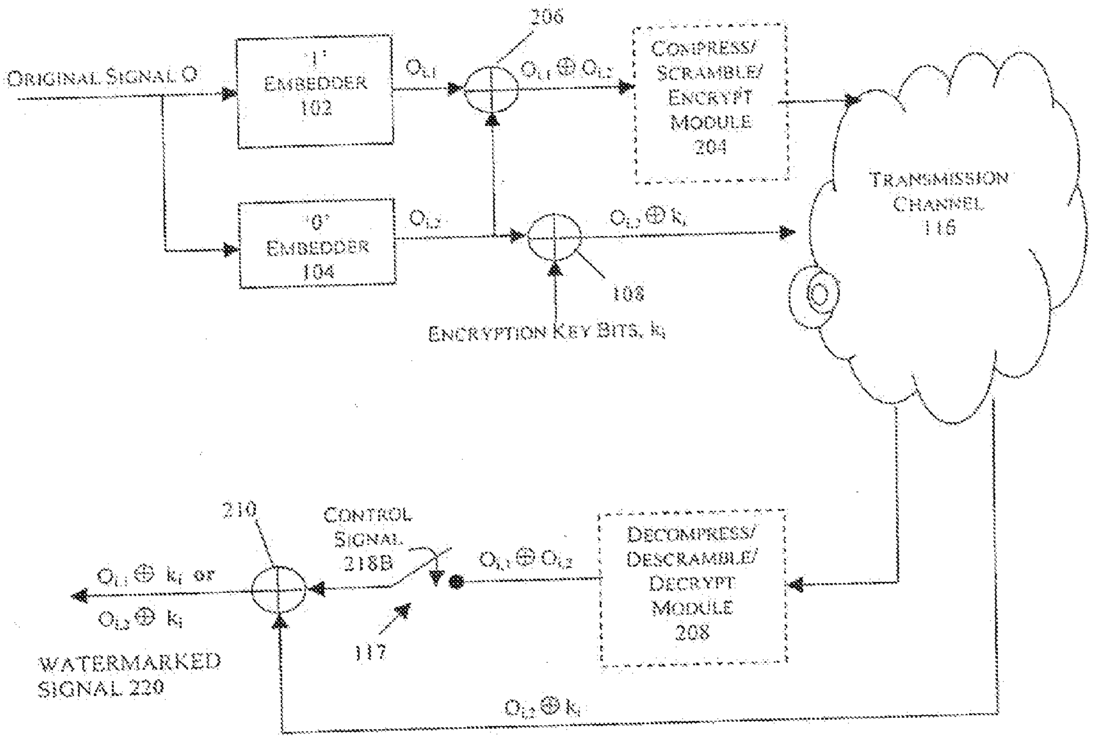
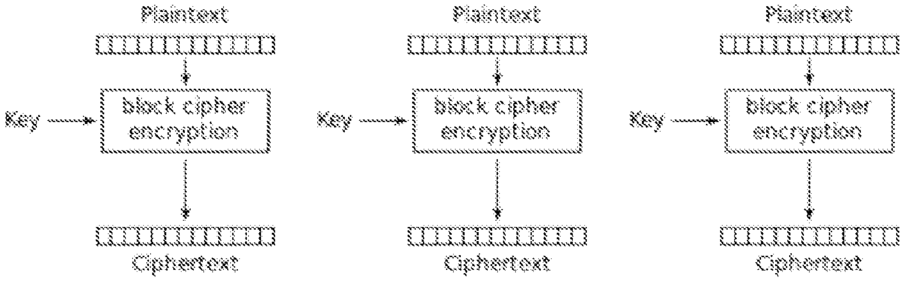
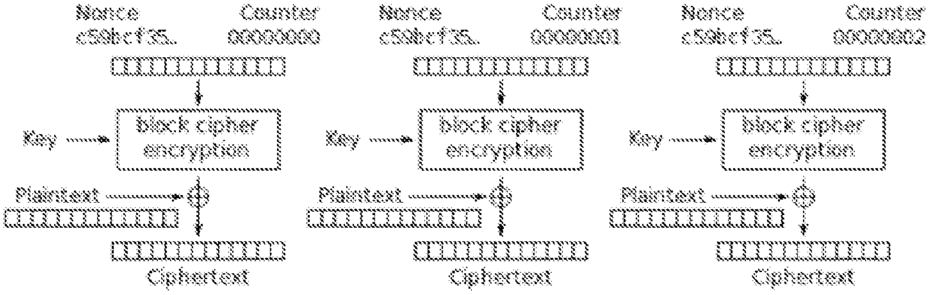

Original Signal O
'1' Embedder 102
'0' Embedder 104
O_{L1}
O_{L2}
206
$O_{L1} \oplus O_{L2}$
Compress/ Scramble/ Encrypt Module 204
$O_{L2} \oplus k_i$
108
Encryption Key Bits, k_i
Transmission Channel 116
Decompress/ Descramble/ Decrypt Module 208
$O_{L1} \oplus O_{L2}$
Control Signal 218B
117
210
$O_{L1} \oplus k_i$ or
$O_{L2} \oplus k_i$
Watermarked Signal 220
$O_{L2} \oplus k_i$

Fig. 2B



Electronic Codebook (ECB) mode encryption

Fig. 3A
(PRIOR ART)



Counter (CTR) mode encryption

Fig. 3B
(PRIOR ART)

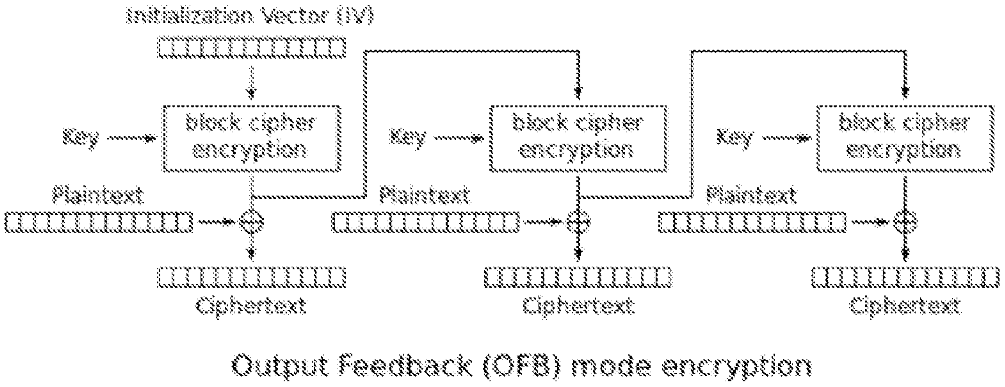


Fig. 3C
(PRIOR ART)

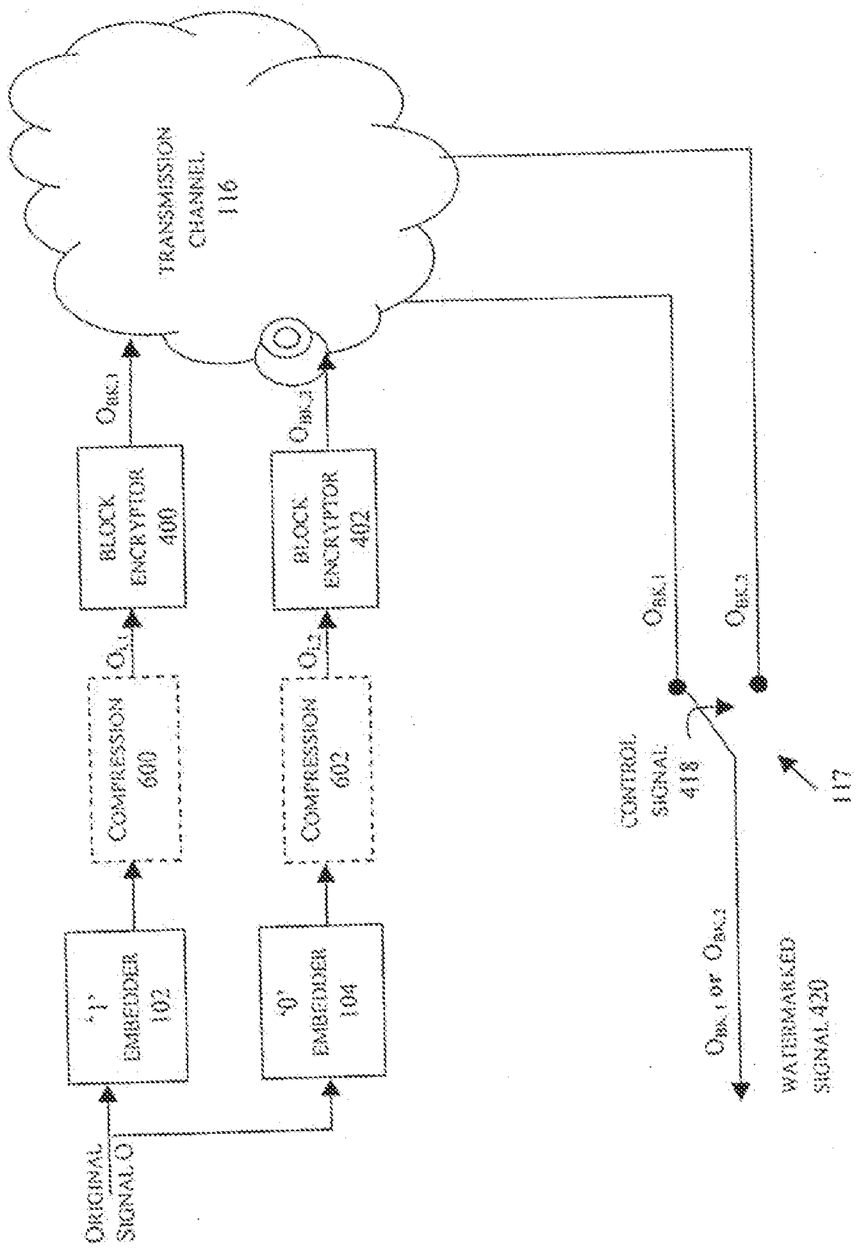

ORIGINAL SIGNAL O
'1' EMBEDDER 102
'0' EMBEDDER 104
COMPRESSION 600
COMPRESSION 602
O_{L1}
O_{L2}
BLOCK ENCRYPTOR 400
BLOCK ENCRYPTOR 402
$O_{BK,1}$
$O_{BK,2}$
TRANSMISSION CHANNEL 116
$O_{BK,1}$
$O_{BK,2}$
CONTROL SIGNAL 418
117
$O_{BK,1}$ or $O_{BK,2}$
WATERMARKED SIGNAL 420

Fig. 4

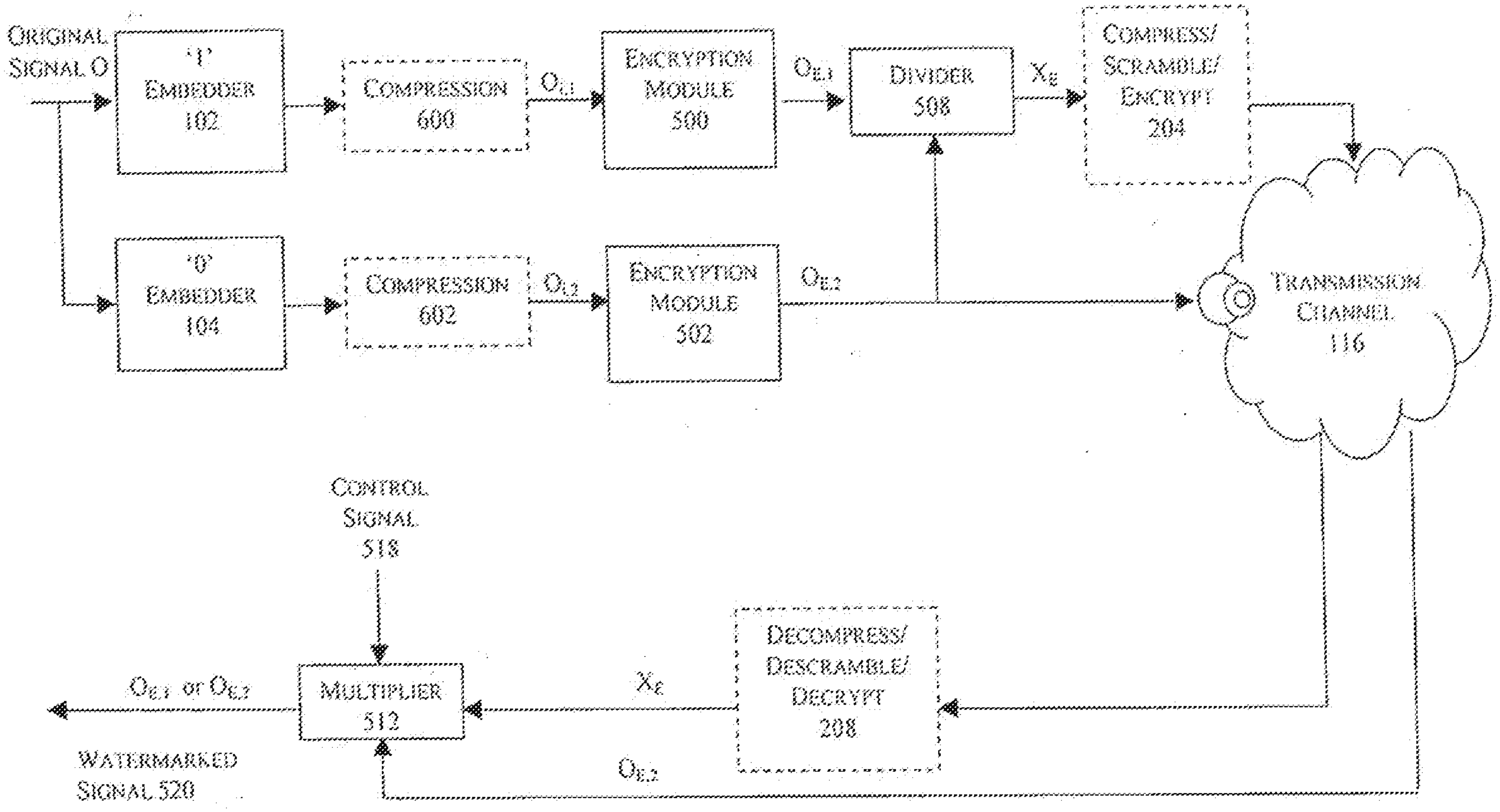

ORIGINAL SIGNAL O
'1' EMBEDDER 102
'0' EMBEDDER 104
COMPRESSION 600
COMPRESSION 602
O_{L1}
O_{L2}
ENCRYPTION MODULE 500
ENCRYPTION MODULE 502
$O_{E,1}$
$O_{E,2}$
DIVIDER 508
X_E
COMPRESS/ SCRAMBLE/ ENCRYPT 204
TRANSMISSION CHANNEL 116
DECOMPRESS/ DESCRAMBLE/ DECRYPT 208
X_E
$O_{E,2}$
CONTROL SIGNAL 518
MULTIPLIER 512
$O_{E,1}$ or $O_{E,2}$
WATERMARKED SIGNAL 520

FIG. 5

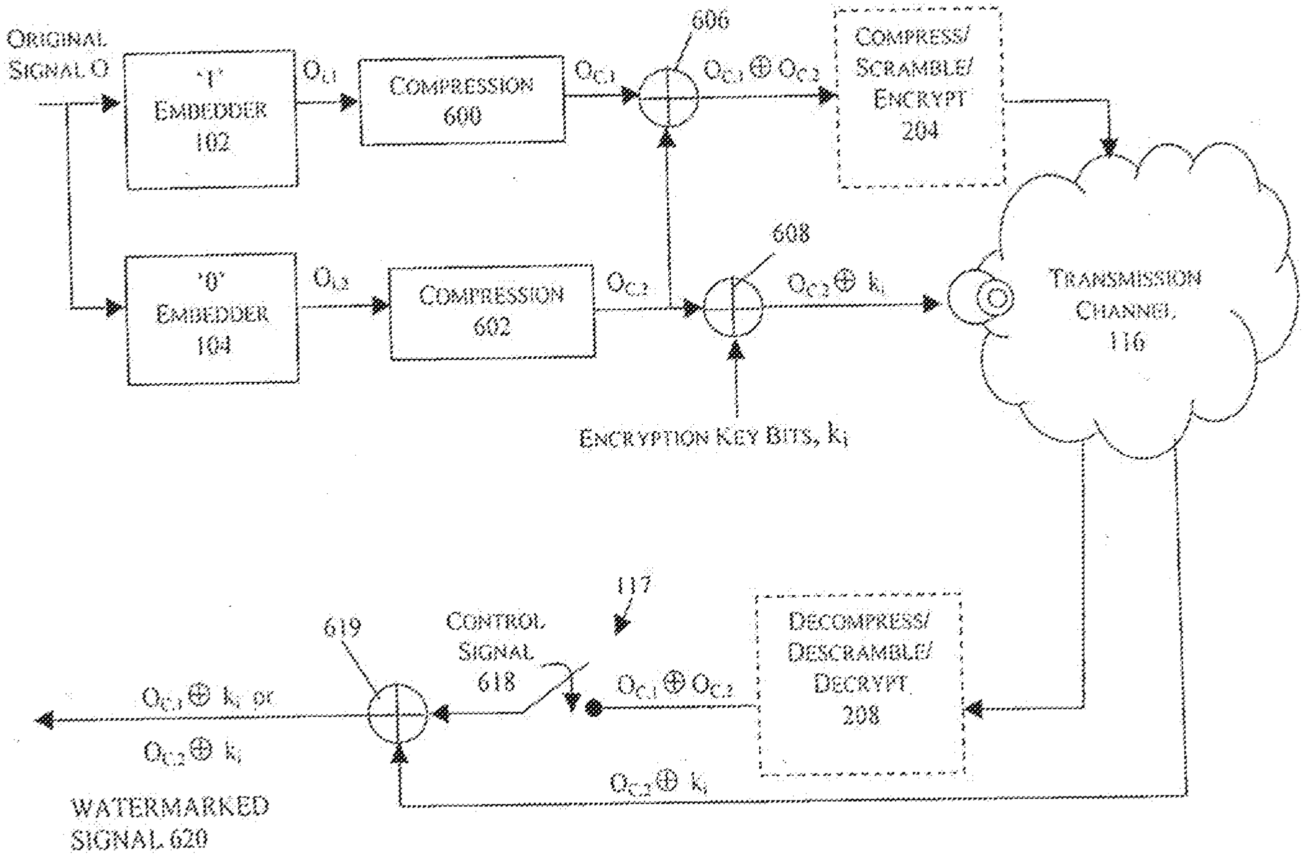

ORIGINAL SIGNAL O
'1' EMBEDDER 102
'0' EMBEDDER 104
O_{L1}
O_{L2}
COMPRESSION 600
COMPRESSION 602
O_{C1}
O_{C2}
606
$O_{C1} \oplus O_{C2}$
COMPRESS/ SCRAMBLE/ ENCRYPT 204
608
$O_{C2} \oplus k_i$
ENCRYPTION KEY BITS, k_i
TRANSMISSION CHANNEL 116
DECOMPRESS/ DESCRAMBLE/ DECRYPT 208
117
CONTROL SIGNAL 618
$O_{C1} \oplus O_{C2}$
619
$O_{C2} \oplus k_i$
$O_{C1} \oplus k_i$ or
$O_{C2} \oplus k_i$
WATERMARKED SIGNAL 620

FIG. 6

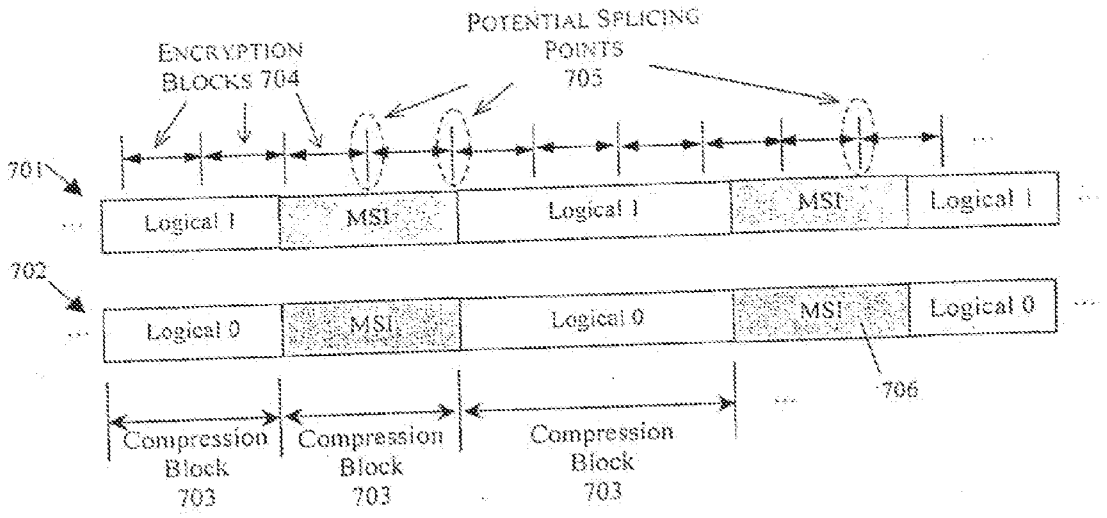

POTENTIAL SPLICING POINTS 705
ENCRYPTION BLOCKS 704
701
702
Logical 1
MSI
Logical 1
MSI
Logical 1
Logical 0
MSI
Logical 0
MSI
Logical 0
706
Compression Block 703
Compression Block 703
Compression Block 703

FIG. 7

WATERMARKING IN AN ENCRYPTED DOMAIN

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of and claims priority to U.S. patent application Ser. No. 11/482,519, filed on Jul. 7, 2006, which claims priority to U.S. Provisional Patent Application No. 60/697,515 filed on Jul. 7, 2005, which are incorporated herein by reference for all purposes in their entirety.

BACKGROUND OF THE INVENTION

[0002] Encryption techniques are often utilized to protect multimedia content signals during their storage or transport from one location to the next. The encrypted content may be securely broadcast over the air, through the Internet, over cable networks, over wireless networks, distributed via storage media, or disseminated through other means with little concern about piracy of the content. The level of security of the encrypted content depends on, among other things, the strength of the encryption algorithm and the encryption key management and safekeeping.

[0003] Before describing the details of the present invention it is beneficial to review some common encryption algorithms and techniques. More detailed descriptions may be found in, for example, "Applied Cryptography" by B. Schneier (John Wiley & Sons: New York, 1996; ISBN: 0-471-12845-7). One class of encryption algorithms, called Stream Ciphers, converts the unencrypted content into an encrypted ciphertext one bit at a time. In this case, the content (i.e., the plaintext) is treated as a stream of bits, p_i, that are XORed with a stream of encryption key bits, k_i, to produce the encrypted (i.e., ciphertext) bits, c_i. Equation (1) describes this process mathematically:

$$cp_i=p_i\oplus k_i \qquad \text{Equation (1)}$$

[0004] The encryption key bits, k_i, are typically generated independently using key stream generators known in the art. At the decryption end, the encrypted stream is XORed with an identical key stream to produce the original content. The decryption operation is mathematically represented by Equation (2).

$$p_i=(p_i\oplus k_i)\oplus k_i \qquad \text{Equation (2)}$$

[0005] In another class of encryption algorithms, called Block Ciphers, the content is processed in blocks of fixed size. So for example, a digital content may first be parsed into blocks of 64 bits and then each 64-bit block may be encrypted according to the encryption algorithm. Some of the most widely used encryption algorithms such as DES and AES are block ciphers. Block ciphers may further operate in different modes. In particular, in Electronic Codebook (ECB) and Counter (CTR) modes of operation, each block is encrypted independently from other blocks in the content. In Cipher Block Chaining (CBC) mode, Output Feedback (OFB) mode and Cipher Feedback (CFB) mode, each encrypted block has a dependency on the neighboring ciphertext and/or plaintext blocks. Cryptographic algorithms may also be classified as symmetric or asymmetric algorithms. In symmetric algorithms the same key is used for encryption and decryption, whereas in asymmetric algorithms different keys, and possibly different algorithmic steps, are used for encryption and decryption of the content.

[0006] While access to an encrypted content may be limited to entities with proper authorization and decryption keys, once a content is decrypted, it may be readily copied and disseminated. This is particularly true for multimedia content that must inevitably be converted to audio and/or visual signals (e.g., analog format) in order to reach an audience. Watermarks are particularly well suited to plug this so-called 'analog hole'. Digital watermarking is typically referred to as the insertion of auxiliary information bits into a host signal without producing perceptible artifacts. Watermark bits embedded into a host signal are designed to be imperceptible, robust to common content transformations, and resistant to intentional attacks that are targeted to remove or alter the watermarks. The detection of watermarks as well as the extraction of information carried in the watermarks may be used to trigger a variety of actions and enable a myriad of applications. Some of these applications include copy control, broadcast monitoring, rights management, authentication and integrity verification, forensic tracking and covert communication. Numerous watermarking algorithms and applications are described in the prior art.

[0007] Due to the complimentary roles of digital watermarking and encryption in the safekeeping and management of content, both techniques are often used to protect and manage content of significant value such as audio, video, still images, text, programming data and other information in digital or analog formats. In an example workflow of content preparation and distribution, a content may be first embedded with digital watermarks; then it may optionally be compressed (to save storage space and/or transmission bandwidth) and finally, it may be encrypted prior to being transmitted or stored outside of a secure environment. Note, that in some applications, the insertion of watermarks may alternatively, or additionally, take place after the compression of the content but prior to the encryption. In some applications, however, it may be advantageous to insert digital watermarks directly into an encrypted data stream (without first decrypting the content). For example, in a forensic tracking application, a digital movie, after appropriate post production processing, may be encrypted at the movie studio or post production house, and sent out for distribution to movie theatres, to on-line retailers, or directly to the consumer. In such applications, it is often desired to insert forensic or transactional watermarks into the movie content to identify each entity or node in the distribution channel, including the purchasers of the content, the various distributors of the content, the presentation venue and the time/date/location of each presentation or purchase. Since a multiplicity of purchase/presentation requests may be received at any given time, it is also desired to insert the watermarks expeditiously and efficiently into the content without introducing significant delays in the processing and transmission of the requested content.

[0008] One way to achieve this goal would be to, at each desired node of the distribution channel, decrypt and possibly decompress the content, insert the appropriate watermarks and then re-compress and re-encrypt the embedded content. This procedure not only requires the knowledge of the encryption/decryption algorithms as well as the presence of encryption/decryption keys at each distribution node, but is also likely to introduce significant delays in the processing of the content. While it may be possible to securely communicate the encryption/decryption keys to theses nodes and produce a secure environment for the encryption/decryption to take place, this task would require additional system design, network security operations and key management protocols which may affect the operational cost and overall security of the distribution system.

[0009] It would be advantageous to provide methods, apparatus, and systems for digital watermarking that overcome various deficiencies of the prior art by providing the capability of watermark insertion into an encrypted content signal. In particular, it would be advantageous to provide methods, apparatus, and systems for the insertion of watermarks into an encrypted digital content that do not require the decryption and subsequent re-encryption of the digital content. It would also be advantageous to allow secure insertion of digital watermarks at any point in the transmission, storage or distribution of an encrypted digital content, without the need to decrypt (and further re-encrypt) the encrypted digital host content signal, and without requiring the knowledge of the encryption/decryption keys. It would be further advantageous if such embedded watermarks were adapted to persist throughout the content after it has undergone decryption. It would be still further advantageous to enable the insertion of digital watermarks into an encrypted host content that is in a compressed format and in such a way that the embedded watermarks persist throughout the content even after decryption and decompression of the host content signal. It would be advantageous if such techniques were applicable to a host content that has been encrypted using a variety of different encryption techniques, including stream ciphers, block cipher, symmetric and asymmetric encryption algorithms.

[0010] The methods, apparatus, and systems of the present invention provide the foregoing and other advantages.

SUMMARY OF THE INVENTION

[0011] The present invention provides methods, apparatus, and systems for the insertion of watermarks into an encrypted digital content that do not require decryption and subsequent re-encryption of the content.

[0012] In one example embodiment of the present invention, a method for embedding auxiliary information symbols in an encrypted host content signal is provided. A first version of a host content signal embedded with a first logical value is encrypted to produce a first encrypted signal. A second version of the host content signal embedded with a second logical value is encrypted to produce a second encrypted signal. A first set of segments from the first encrypted signal is combined with a second set of segments from the second encrypted signal in a pre-defined manner to produce a composite encrypted host content with embedded auxiliary information.

[0013] The first and second encrypted signals may be in a compressed format. In particular, the two versions of the host content signals embedded with respective first and second logical values may be in a compressed format prior to encryption. Alternatively, they may be compressed after encryption and before transmission to a client device or user location.

[0014] The combining of the segments from the encrypted signals may be performed without the use of the encryption or decryption keys.

[0015] The embedded auxiliary information may persist throughout the host content after decryption of the composite encrypted host content.

[0016] The encrypting of the first and second versions of the host content signal may occur at a pre-processing center and the combining may occur at a user location.

[0017] A further example embodiment of the present invention provides a method for embedding auxiliary information symbols in an encrypted host content signal. A first version of an original host content signal embedded with a first logical value is encrypted to produce a first encrypted signal. A second signal comprising information corresponding to the first logical value and a second logical value embedded in the host content signal is produced. A first set of segments from the first encrypted signal is combined with a second set of segments from the second signal in a pre-defined manner to produce a composite encrypted host content with embedded auxiliary information.

[0018] The method may further include at least one of compressing, encrypting, and scrambling the second signal.

[0019] The host content signal may be in a compressed format. For example, the encrypting may comprise encrypting of the compressed host content signal embedded with a first logical value. Further, the second signal may comprise information corresponding to the first and second logical values embedded in the compressed host content signal.

[0020] A further example embodiment of a method for embedding auxiliary information symbols in an encrypted host content signal in accordance with the present invention is provided. The host content signal is encrypted to produce an unmarked encrypted host content signal. A first signal is produced which comprises information corresponding to a first logical value embedded in the host content signal. A second signal is produced comprising information corresponding to a second logical value embedded in the host content signal. A first set of segments is selected from the first signal and a second set of segments are selected from the second signal. The first set and the second set of segments are combined with the unmarked encrypted host content in a predefined manner to produce a composite encrypted host content with embedded auxiliary information.

[0021] The method may further include at least one of compressing, encrypting, and scrambling the first or second signal.

[0022] The host content signal may be in compressed format. In such an embodiment, the encrypting may comprise encrypting of the compressed host content signal. Similarly, the first and second signals may comprise information corresponding to first and second logical values embedded in the compressed host content signal, respectively.

[0023] The first and second signals may be transmitted to a user premises, and combined with the unmarked encrypted host content signal that resides at the user premises.

[0024] In a further example embodiment of the present invention, a method for embedding auxiliary information symbols in a compressed and encrypted host content signal is provided. A first version of a compressed host content signal embedded with a first logical value is encrypted to produce a first encrypted signal. A second signal is produced which comprises information corresponding to the first logical value and a second logical value embedded in the compressed host content signal. A first set of segments from the first encrypted signal is combined with a second set of segments from the second signal in a pre-defined manner to produce a composite encrypted host content with embedded auxiliary information.

[0025] The first encrypted signal and the second signal may comprise a matching signal interval. The combining may occur within the matching signal interval.

[0026] The first encrypted signal may comprise a partially encrypted signal.

[0027] Additional meta data corresponding to the first encrypted signal and the second signal are produced to facilitate the combining of the segments.

[0028] An example embodiment of the present invention also includes a further method for embedding auxiliary information into an encrypted host signal. In this embodiment, a first signal comprising an encrypted first version of a host signal is received, for example at a client device or user location. A second signal comprising information related to a first and a second logical values embedded in a second version of the host signal is also received. At least portions of the second signal are then combined with the first signal in a pre-defined manner to produce a composite encrypted host signal with embedded auxiliary information.

[0029] An additional method for embedding auxiliary information into an encrypted host signal in accordance with an example embodiment the present invention is also provided. In this embodiment, an information signal corresponding to first and second logical values embedded into a first version of the host signal is received (e.g., at a client device or user location) from a pre-processing center. This information signal may then be combined with an encrypted second version of the host signal in a pre-defined manner to produce a composite encrypted host signal with embedded auxiliary information. This encrypted version of the host signal may already be present at the client device or user location, or received thereat simultaneously with the information signal.

[0030] In any of the foregoing example embodiments, the encrypted signal may be encrypted in accordance with at least one of a stream cipher, a block cipher, a symmetric encryption algorithm, an asymmetric encryption algorithm, or the like. Further, the predefined manner of the combining of the segments may identify an entity or a transaction. In addition, the host content signal may comprise at least one of audio, video, text, and programming information.

BRIEF DESCRIPTION OF THE DRAWINGS

[0031] The present invention will hereinafter be described in conjunction with the appended drawing figures, wherein like reference numerals denote like elements, and:

[0032] FIG. **1** is a block diagram showing the insertion of watermarks into a content encrypted with a stream cipher in accordance with an example embodiment of the invention;

[0033] FIG. **2**A is a block diagram showing the insertion of watermarks into a content encrypted with a stream cipher in accordance with an example embodiment of the invention;

[0034] FIG. **2**B is a block diagram showing the insertion of watermarks into a content encrypted with a stream cipher in accordance with an example embodiment of the invention;

[0035] FIG. **3**A illustrates an Electronic Codebook (ECB) block encryption scheme;

[0036] FIG. **3**B illustrates a Counter (CTR) block encryption scheme;

[0037] FIG. **3**C illustrates an Output Feedback (OFB) block encryption scheme;

[0038] FIG. **4** is a block diagram showing the insertion of watermarks into a content encrypted with a block cipher in accordance with an example embodiment of the invention;

[0039] FIG. **5** is a block diagram showing the insertion of watermarks in an RSA-like encrypted content in accordance with an example embodiment of the present invention;

[0040] FIG. **6** is a block diagram showing the insertion of watermarks into a compressed and encrypted content in accordance with an example embodiment of the invention;

[0041] FIG. **7** shows the insertion of specially tailored watermarks into a compressed and encrypted content in accordance with an example embodiment of the invention.

DETAILED DESCRIPTION

[0042] The ensuing detailed description provides exemplary embodiments only, and is not intended to limit the scope, applicability, or configuration of the invention. Rather, the ensuing detailed description of the exemplary embodiments will provide those skilled in the art with an enabling description for implementing an embodiment of the invention. It should be understood that various changes may be made in the function and arrangement of elements without departing from the spirit and scope of the invention as set forth in the appended claims.

[0043] Several techniques for the insertion of forensic or transactional watermarks have been previously described in the literature. Some of these techniques take advantage of the fact that computationally expensive operations of the embedding **1**, process may be carried out separately, at a pre-processing center, prior to the embedding of watermarks. Once a request for the delivery of a content is received, the pre-processed versions of the content signal may be combined, without requiring computationally expensive operations, to produce a content with embedded watermarks. For example, in accordance with commonly owned U.S. Pat. No. 6,912,315, adding forensic or transactional watermarks may be accomplished by pre-processing an original content with two or more different logical values to produce two or more embedded content signals. The two or more embedded content signals may then be transmitted to the 'client' (e.g., to an on-line distribution center or to a user premises), where the appropriate portions of the preprocessed signals are selected and assembled together to form an embedded content with a desired watermark value. Other variations and improvements to this technique are disclosed in commonly owned co-pending U.S. patent application Ser. No. 11/124,465. These improvements produce a versatile watermarking system that requires a smaller bandwidth for the transmission and storage of reduced-scale signals that enable the insertion of forensic marks. The term reduced-scale signal (as opposed to full-scale signal) is used to refer to any signal with a smaller information content than the original content. For example, such signals may have a smaller duration, dynamic range, bandwidth and/or spatial resolution than the original content.

[0044] Alternatively, as disclosed commonly owned U.S. Pat. No. 6,430,301, each of two pre-processed versions of the content comprises two separate regions. The first region, called the Matching Signal Interval (MSI), either contains no watermark value or is embedded identically with the same watermark value in both pre-processed versions of the content (this is referred to as the region with 'common watermark'). The second region, which is time interleaved with the first region, is embedded with a first or a second logical value, in the first or second pre-processed versions of the content, respectively. Transactional watermarking may be implemented by assembling proper portions of the first version of the content with proper portions of the second version of the content to produce an embedded content. The cutting and splicing of the two versions all occur within the regions, where two signals are identical.

[0045] The various embodiments of the present invention enable the insertion of watermarks, such as the ones produced by the aforementioned watermarking systems, into an encrypted data stream. The disclosed methods and systems are applicable to systems that utilize stream ciphers, block ciphers, and symmetric or asymmetric encryption algorithms. In accordance with further embodiments of the present invention, such watermarks may be embedded into “compressed-and-encrypted’ data signals. This is accomplished without requiring the content to be decrypted or decompressed, and without any knowledge of the encryption or decryption keys. Although most of the example embodiments of the present invention are described by illustrative examples that involve the embedding of binary watermark values, it should be understood that these techniques are readily extended to include the embedding of non-binary data symbols or embedding of multiple layers of watermarks (e.g., such as disclosed in the commonly owned U.S. Pat. No. 6,912,315, and commonly owned co-pending U.S. patent application Ser. No. 11/124,465) that can be embedded and extracted independently from one another.

Stream Ciphers

[0046] The particular methodology used to achieve the various goals of the present invention depends on the type of encryption algorithm and other configuration parameters of the media delivery and preparation system. The following describes the insertion of watermarks in example embodiments of a system that utilizes stream ciphers.

[0047] Case 0: This is the default case, where the original content signal, and/or a set of signals containing embedded logical values, is encrypted and transmitted to the target destination in accordance with an example embodiment of the invention. The received signals are then decrypted and appropriately cut-and-spliced, in accordance with any one of the above-described prior art transactional watermarking techniques, to produce a content signal with a desired watermark value. The embedded content may then be optionally re-encrypted and transmitted to the next destination. The re-encryption is typically not necessary if the content watermarking and subsequent transmission are conducted all within a secure environment. This technique is equally applicable to all types of encryption algorithms and different variations of forensic/transactional watermarking since the watermark is applied to the plaintext signal.

[0048] Case 1: In this example, illustrated in FIG. **1**, two full-scale versions of the original content O signal are generated at the pre-processing stage, by embedding a first logical value in a first version of the original content signal O at embedder **102** and embedding a second logical value in a second version of the original content O at embedder **104**. The term “full-scale” refers to a signal that is substantially similar to the original content signal as described in the pending U.S. patent application Ser. No. 11/124,465. In FIG. **1**, embedder **102** is shown as embedding a logical “1” in the first version of the original content signal O and embedder **104** is shown as embedding a logical “0” in the second version of the original content signal O. However, those skilled in the art will appreciate that embedders **102**, **104** may both be capable of embedding either logical is or Os in the original content signal O.

[0049] Each full-scale version $O_{i,1}$, $O_{i,2}$ is then encrypted and transmitted to the client. This operation may be better understood by examining the following equations. The first full-scale encrypted stream, which is embedded with a first logical value, may be represented by:

$$c_{i,1}=O_{i,1}\oplus k_i, \qquad \text{Equation (3),}$$

where, i represents the bit position within the stream of bits in the first version of the full-scale signal, $O_{i,1}$ represents the unencrypted version of the first full-scale signal at position i, and $c_{i,1}$ represents the encrypted version of the first full-scale signal at position i. The second full-scale encrypted stream, which is embedded with a second logical value, may be similarly represented by:

$$c_{i,2}=O_{i,2}\oplus k_i \qquad \text{Equation (4),}$$

where, $O_{i,2}$ represents the unencrypted version of the second full-scale signal at position i, and $c_{i,2}$ represents the encrypted version of the second full-scale signal at position i. The encryption process occurs on a bit-by-bit basis and may comprise XORing each version with key bit stream, k_i at XOR operators **106** and **108**, respectively, to produce the two full scale encrypted streams $O_{i,1}\oplus k_i$ and $O_{i,2}\oplus k_i$. The encryption of the two versions $O_{i,1}$, $O_{i,2}$ must occur independently from one another, but synchronously with the same encryption key stream k_i. The two full scale encrypted streams $O_{i,1}\oplus k_i$ and $O_{i,2}\oplus k_i$ may then be transmitted over transmission channel **116** to the client side (user location or user device). At the client side, the desired portions of one stream (e.g., stream $O_{i,1}\oplus k_i$,) may be combined (on a bit-by-bit basis) with the desired portions of the other stream (e.g., $O_{i,2}\oplus k_i$) in accordance with a control signal **118** to produce a composite encrypted stream **120** with embedded watermarks.

[0050] Since the two versions are encrypted synchronously with the same key bit stream, k, the bits of one encrypted stream $O_{i,1}\oplus k_i$ may replace the corresponding bits of the other stream $O_{i,2}\oplus k_i$, to form a composite data stream (watermarked signal **120**). The composite signal **120** maintains its encryption and may be fully decrypted using the same encryption key stream, k_i.

[0051] In FIG. **1**, and in the remainder of this disclosure, the term ‘control signal’ is used to generically represent any combination of instructions, timing information, logical values or other signals that enable the assembly of particular segments of the two or more versions of the content signal. In a simple example involving the insertion of transactional watermarks in accordance with FIG. **1**, a particular sequence of bits that identify the purchaser of a content may be required to be embedded into the content. The control signal **118**, in this example, may simply enable the switching between the two encrypted streams at watermark bit boundary locations (e.g., using switch **117** controlled by control signal **118**). For example, if the watermarking scheme calls for watermarking bits that span 100 samples of the host content signal, the switching occurs at content bit locations **101**, **201**, **301**, ... Obviously, if the present watermark bit has the same value as the previous watermark bit, no switching needs to occur for the present bit duration. In other examples, the generation of the control signal **118** may include more complicated operations. These operations may involve the generation of auxiliary information that comprise watermark payload (e.g., the generation of a time stamp from local clock), the generation of synchronization sequences, the generation or selection of bit transition functions, the application of various channel coding techniques, such as error correction codes, and other necessary operations to produce a stream of logical values that are subsequently embedded into the content.

[0052] The transmission channel **116** shown in FIG. **1** represents any one or more of a variety of communication channels that may be used to transmit or store information. Examples of such communication channels include, but are not limited to, the Internet, local area networks, wide area networks, satellite and over-the-air broadcast channels, magnetic, optical or electronic storage devices, and the like.

[0053] Case 2A: In the example embodiment shown in FIG. **2**A, three signals are transmitted to the client side. One full-scale signal is a version of the original host content signal O. This version of the original host content signal O_i is encrypted on a bit-by-bit basis at XOR operator **110** to produce encrypted stream $O_i \oplus k_i$. The other two signals are produced by first embedding two logical values in separate versions of the original content signal O (at embedders **102** and **104**) to produce two embedded signals $O_{i,1}$ and $O_{i,2}$ as discussed above in connection with FIG. **1**. These signals $O_{i,1}$ and $O_{i,2}$ are next XORed with the original signal O_i at XOR operators **200**, **202**, respectively to produce signals $O_{i,1} \oplus O_i$ and $O_{i,2} \oplus O_i$. Signals $O_{i,1} \oplus O_i$ and $O_{i,2} \oplus O_i$ are optionally compressed, scrambled or encrypted (e.g., at compress/scramble/encrypt module **204**) prior to their transmission over transmission channel **116** to the client side (e.g., a user location or user device).

[0054] These optional operations at module **204** may be necessary to reduce the transmission bandwidth and to enhance the security of the transmitted signals. The information content of the generated signals, $O_{i,1} \oplus O_i$ and $O_{i,2} \oplus O_i$, is typically substantially smaller than the original content signal since these signals are produced by XORing two substantially similar signals (recall that XOR operation produces a '1' value only if the two operands are different). Thus the signals generated by XOR operations may comprise many zeroes, a property that makes them a good candidate for the application of lossless compression techniques. In other cases, where the embedded and original signals contain large differences (for example, as a result of applying watermark masking/concealment techniques during the embedding process), such compression techniques may not be as effective.

[0055] Upon the reception and appropriate decompression, descrambling or decryption at module **208**, appropriate portions of the signals $O_{i,1} \oplus O_i$ and $O_{i,2} \oplus O_i$ may be XORed (e.g., at XOR operator **210**) with the original encrypted content, $O_{i,1} \oplus k_i$, in accordance with the control signal **218**A, producing a final composite encrypted watermarked signal **220**, portions of which contain the first embedded watermark value (e.g., a logical "1") and portions of which contain the second embedded watermark value e.g., a logical "0").

[0056] One of the features of the watermarking technique described in FIG. **2**A is its ability to decouple watermarking and encryption operations. In other words, the embedders **102**, **104** and XOR operators **200**, **202** in this architecture do not need access to the encryption keys at all. In addition, the original signal O remains intact and can be independently transmitted to other destinations that do not require (or perhaps forbid) the presence of embedded watermarks. Using this technique, it is also possible to produce the appropriate watermark signals for a content that is already at a user premises. In this case, if an exact copy of the original content signal O is available at the pre-processing center, the signals $O_{i,1} \oplus O_i$ and $O_{i,2} \oplus O_i$ may be generated and transmitted to the user. This allows dynamic modifications of watermarking techniques and parameters that can be subsequently transmitted to the client for insertion of the watermark.

[0057] Case 2B: The example embodiment shown in FIG. **2**B is similar to Case 2A above, with the exception that only two signals are generated and transmitted to the desired destination. The two streams, $O_{i,1}$ and $O_{i,2}$, are produced by embedding separate versions of the original content signal O with the first and second logical values at embedders **102**, **104** as discussed above in connection with FIGS. **1** and **2**A. These signals, $O_{i,1}$ and $O_{i,2}$ and are XORed with each other at XOR operator **206** to produce the signal $O_{i,1} \oplus O_{i,2}$. Similar to the reasoning discussed above in Case 2A, this signal may contain a larger proportion of zero-valued bits and may be a good candidate for the application of a lossless compression technique. Further scrambling and encryption operations may also be applied (e.g., at module **204**) to protect this signal prior to transmission to the client over transmission channel **116**. Appropriate decompression, descrambling or decryption may be applied as necessary to signal $O_{i,1} \oplus O_{i,2}$ at module **208**.

[0058] One of the embedded content signals (i.e., the signal $O_{i,2}$ in the example embodiment of FIG. **2**B) is encrypted on a bit-by-bit basis with encryption key stream k_i at XOR operator **108** to produce $O_{i,2} \oplus k_i$ which is also transmitted to the client side over transmission channel **116**. On the reception side, the signal $O_{i,1} \oplus O_{i,2}$ may be XORed with the signal $O_{i,2} \oplus k_i$ at XOR operator **210** in accordance with a control signal **218**B to produce an encrypted watermarked content signal **220** with appropriate embedded watermark values. In the example embodiment of FIG. **2**B, when the switch **117** is in open position, the encrypted signal $O_{i,2} \oplus k_i$ appears unchanged at the output of the XOR operator **210**. When in accordance with the control signal **218**B, the switch **117** is flipped to a closed position, a portion of the signal $O_{i,1} \oplus O_{i,2}$ is XORed with a corresponding portion of the encrypted signal $O_{i,2} \oplus k_i$, producing an encrypted segment $O_{i,1} \oplus k_i$ at the output of the XOR operator **210**.

[0059] The example watermarking technique described in connection with FIG. **2**B produces the same embedded content as the that produced by the example watermarking technique described in connection with FIG. **2**A, but it requires a smaller transmission bandwidth. In the architecture of FIG. **2**B, it only suffices to transmit two signals to the client, one of which, namely $O_{i,1} \oplus O_{i,2}$, is a reduced-scale signal (or can be converted to a reduced-scale signal).

Block Ciphers

[0060] The above described watermark insertion techniques described in the context of stream ciphers may be readily adapted to operate with block encryption algorithms, FIGS. **3**A, **3**B and **3**C provide schematic diagrams of three typical block encryption modes of operation that are well known in the art. In Electronic Code Book (ECB) mode of FIG. **3**A, each block of content data (plaintext) is replaced by an alternate block of encrypted data of the same size (ciphertext). In the Counter (CTR) mode of operation shown in FIG. **39**, encryption key blocks (ciphertext blocks) are generated for each data block (plaintext) by encrypting successive values of a "counter". The counter can be any simple function which produces a sequence which is guaranteed not to repeat for a tong time. In Output Feedback (OFB) mode of operation shown in FIG. **3**C, an initialization vector (IV) (which can be a random number) is used to generate the encryption key for the first block of plaintext; this key is used to generate other encryption keys for the subsequent blocks of plaintext. In both the CTR and OFB configurations, the generated key blocks are XORed with the plaintext blocks to form the block encrypted content signal.

[0061] The insertion of watermarks into such block encrypted data streams can be done by adjusting a few watermark design parameters. One such adjustment would be to select the watermark bit durations to be an integer multiple of cipher block size. For example, an audio stream with 16-bit sample values and a block cipher size of 128 bits may be used. The watermarking system may be designed to embed a single watermark bit into 440 samples of the audio content (this corresponds to a watermark bit rate of approximately 100 bits per second for a 44.1 KHz audio signal). Thus 440×16=7040 bits of the host signal would be required for the embedding of each watermark bit. If the audio signal were encrypted in blocks of 128 bits, then 740/128=55 cipher blocks would be required to carry each watermark bit. An exemplary procedure would involve the embedding of the host signal with two different logical values to produce two embedded signals, wherein each embedded logical value spans 440 samples of the host signal. Each of the two embedded signals produced this way may then be grouped into 128-bit blocks, encrypted, and transmitted to the desired destination. The encryption must take place synchronously for both versions of the embedded content signal with the same encryption key. FIG. **4** describes an exemplary embodiment of this procedure. The '1' and '0' embedding modules **102**, **104** produce two embedded signals $O_{i,1}$ and $O_{i,2}$ from the original content signal O such that each embedded watermark symbol spans one or more full encryption blocks. The embedded signals, $O_{i,1}$ and $O_{i,2}$, are next encrypted by the block encryptor modules **400**, **402** to generate encrypted signals $O_{BK,1}$ and $O_{BK,2}$, which can then be transmitted to the client via transmission channel **116**. In FIG. **4**, the subscript B designates the block processed signal samples and the subscript K is used to designate such blocks that are encrypted. Upon the reception of encrypted block signals $O_{BK,1}$ and $O_{BK,2}$, appropriate portions of these signals are selected in accordance with the control signal **418**, and assembled together to produce an embedded content signal **420**. Using the exemplary numerical values discussed earlier, the control signal **418** must switch between the two signals at multiples of 55 encryption blocks (i.e., **1** embedded watermark bit) or 740 encrypted bits.

[0062] The above technique may be modified to conform to other encryption block sizes, sampling rates or bit depth values. These modifications may produce different watermark bit rates in order to make the duration of each watermark bit an integer multiple of encryption block size. For example, if the bit depth of the incoming audio signal in the above example were to become 32 bits, a watermark bit rate of approximately 200 bits-per-second would be required to produce the same number of blocks per bit (i.e., 55, 128-bit encryption blocks per watermark bit). Alternatively, the number of blocks-per-watermark-bit may be changed from 55 to 110 to maintain the watermark bit rate at ~100 bits-per-second while accommodating 32-bit audio sample values. Furthermore, the extension of the above described technique to other embedding configurations, such as the ones described in FIGS. **2**A and **2**B, is similarly accomplished by replacing the bit-wise operations by block-wise operations.

Asymmetric and Public-Key Algorithms

[0063] The methods and systems of the various embodiments of the present invention can also be used in conjunction with asymmetric encryption algorithms. These algorithms use different keys for encryption and decryption of the content and may involve different algorithmic operations for encryption and decryption processes. It is important to note that the previously disclosed analysis did not require any references or knowledge of decryption keys or decryption algorithms. Thus, these systems and methods may be readily adapted to operate with asymmetric algorithms, as well. In addition, some asymmetric algorithms require modular arithmetic operations, including exponentiation; these operations and the necessary modifications to the watermark insertion techniques of the present invention will be discussed below.

[0064] RSA is one of the most widely used asymmetric encryption algorithms. RSA uses one key, called the public key, for encryption and another key, called the private key, for decryption of the content. The details of RSA encryption algorithm may be found in a many publications such as, B. Schneier's "Applied Cryptography", John Wiley & Sons: New York, 1996; ISBN: 0-471-12845-7. The basic RSA encryption operation can be described by the following equation:

$$c=O^{k}[\text{modulo } n] \qquad \text{Equation (5),}$$

where c is the encrypted data, 0 is the original, unencrypted data, k is the encryption key, and n is an encryption parameter that is a product of two random prime numbers. In a public-key encryption algorithm, k and n are known public parameters. The decryption is carried out according to the Equation 6:

$$O=c^{d}[\text{modulo } n] \qquad \text{Equation (6),}$$

where d is the private key and is only known to authorized parties. The encryption and decryption operations are carried out in modulo-n arithmetic. Modular arithmetic, and various hardware implementations thereof, is well known in the art and is described in many publications such as, David N. Amanor, "Efficient Hardware Architecture for Modular Manipulation", Master's Thesis, Communications and Media Engineering, University of Applied Sciences Offenburg, Germany, February 2005. In order to encrypt a signal, the signal is typically broken up into smaller numerical blocks. The RSA (or similar asymmetric) encryption algorithm can be better illustrated by considering the following numerical example.

[0065] Let's assume O=688232678, n=3337, d=1019 and block size=3 digits. The original signal O may be broken up into blocks of 3 digits, namely:

[0066] O(1)=688,

[0067] O(2)=232, and

[0068] O(3)=678.

Each block may then be encrypted to produce:

[0069] c(1)=1570,

[0070] c(2)=2756, and

[0071] c(3)=2091.

The encrypted message would then be the concatenation of encrypted blocks:

[0072] c=157027562091.

In the watermarking system of the present invention, each pre-processed version of the original content contains one logical value. For example, the embedding process may produce two embedded sample values in the following manner:

[0073] $O_1(1)=698$, $O_2(1)=678$,

[0074] $O_1(2)$ 240, $O_2(2)=212$,

[0075] $O_1(3)=700$, $O_2(3)=670$,

where O_1 and O_2 represent the embedded signals with the first and second logical values, respectively. These signals are subsequently encrypted to produce encrypted signals that are transmitted to the client. The insertion of watermarks into the encrypted stream then becomes identical to the situation that was described above in connection with FIG. **4**. Portions of the received encrypted signals may then be selected in accordance to a control signal and spliced together to form an encrypted content signal with a particular watermark value. Since the encryption occurs in blocks, the same watermark design considerations that were discussed in connection with FIG. **4** are applicable.

[0076] It is also possible to apply the above techniques to produce systems that are analogous to the ones described in FIGS. **2**A and **29**. For illustration purposes, only the configuration of FIG. **2**B will be analyzed since those skilled in the art will appreciate that similar procedures may be adapted and used in conjunction with the system of FIG. **2**A. The basic idea behind this concept is that a first version of the content embedded with a first logical value, that is also encrypted with an RSA-like algorithm, may be converted to a second version of the content embedded with a second logical value, that is also encrypted, using a multiplicative factor. In order to facilitate the understanding of this concept, the encrypted blocks, $O_1(1)=698$ and $O_2(1)=678$, in the above numerical example are examined. The two embedded blocks $O_1(1)$ and $O_1(2)$ are related to one another by a multiplicative factor (i.e., $O_1(1)=(698/678)*O_1(2)$). In modulo arithmetic this relationship is expressed as: O_1 [modulo n]=$X*O_2$. Determination of the factor X involves modulo division, which is well known in the art and will not be described here. The embedding of watermarks in accordance to the present invention may be carried out using modulo arithmetic as illustrated in the example embodiment shown. FIG. **5**. Separate versions of the original content signal O are embedded with logical values at embedders **102** and **104** to produce two embedded content signals, $O_{i,1}$ and $O_{i,2}$. The two embedded content signals, $O_{i,1}$ and $O_{i,2}$, are encrypted (in this context with an RSA-like algorithm) at encryption modules **500**, **502**, respectively, to produce the signals $O_{E,1}$ and $O_{E,2}$. Next, the multiplicative factors, X_E, that relate the two embedded content signals, are determined using divider module **508**. The subscript E is used to indicate that the multiplicative factors are determined for each encryption unit (i.e., units of bits that are encrypted together). This multiplicative factor may be calculated based on specific parameters of the encryption algorithm using modulo arithmetic. For the RSA example described above, only the knowledge of the public parameter n and the block size (i.e., 3 digits) is required at both the pre-processing center (e.g., operations prior to transmission over transmission channel **116**) and at the client premises (e.g., a user location or device capable or carrying out operations occurring after transmission). The multiplicative factors X_E may be compressed, scrambled, or encrypted at module **204** as required prior to transmission over transmission channel **116**. On the reception side, the multiplicative factors, X_E, may be decompressed, descrambled, or decrypted as needed at module **208**. The multiplicative factors X_E may then be multiplied by the second embedded content, $O_{E,2}$, at multiplier **512** in accordance with a control signal **518** to produce an encrypted content **520** with embedded watermark values. The multiplier module **512** may multiply $O_{E,2}$ by either X_E or 1 (i.e., pass $O_{E,2}$ through unchanged) to produce an encrypted output signal that contains appropriate concatenations of $O_{E,1}$ and $O_{E,2}$.

[0077] While the above examples illustrated the application of the present invention to an RSA-like encryption algorithm, the presented techniques are equally applicable to other asymmetric encryption algorithms, including but not limited to, Pohlig-Hellman, Rabin, EIGamal as well as elliptical curve encryption algorithms.

Insertion of Watermarks into Compressed Domain

[0078] The above-described techniques for the insertion of forensic watermarks may be adapted to operate with an original content signal that is in a compressed format. For a majority of compression schemes, data signals are divided into blocks that are subsequently compressed using a variety of techniques. Examples of such compression algorithms include MPEG, PEG, JPEG2000, AAC, AC3, and the like. What is important is for the particular compression technique to operate on blocks of signal content that can be independently compressed and decompressed. In MPEG compression, for example, a Group of Pictures (GOP) may be considered an independent compression block. FIG. **6** shows an example embodiment of the present invention that includes compression. Other component of this figure are similar to the ones described above in connection with FIG. **2**B (except for the "compression" blocks **600**, **602** that compress the embedded versions of the original content).

[0079] As shown in FIG. **6**, separate versions of the original content signal **0** are embedded with logical values at embedders **102** and **104** to produce two embedded content signals, $O_{i,1}$ and $O_{i,2}$. These signals $O_{i,1}$ and $O_{i,2}$ are compressed at compression modules **600**, **602**, respectively. The signals at the output of the two compression modules **600**, **602** are labeled $O_{C,1}$ and $O_{C,2}$, respectively, in order to designate compressed signals that are generated in independent blocks (i.e., compression blocks), In the upper path, signals $O_{C,1}$ and $O_{C,2}$ are XORed together at XOR operator **606** on a compression block-by-block basis to produce a "difference" signal between the two versions of the embedded-and-compressed signals. The difference signal output from XOR operator **606** is designated $O_{C,1}\oplus O_{C,2}$. The difference signal $O_{C,1}\oplus O_{C,2}$ may be compressed, scrambled, or encrypted at module **204** as required prior to transmission over transmission channel **116**, in the lower path, $O_{C,2}$ is XORed with the key sequence, at XOR operator **608**, on a bit-by-bit basis to produce an encrypted stream $O_{C,2}\oplus k_i$. On the reception side, the signal $O_{C,1}\oplus O_{C,2}$ may be decompressed, descrambled, or decrypted as needed at module **208**. One or more blocks of the signal $O_{C,1}\oplus O_{C,2}$ may then be appropriately selected in accordance with a control signal **618** and XORed with the signal $O_{C,2}\oplus k_i$ at XOR operator **619** to produce a compressed-and-encrypted signal **620** with the desired forensic watermark. The operation of the switch **117** of FIG. **6** is similar to the operation described in relation with FIG. **2**B.

[0080] Similarly, the embodiments shown in FIGS. **1**, **2**A, **4**, and **5** may employed where the original content signal is first embedded with the logical values and then compressed, as indicated by the optional compression modules **600** and **602** (shown in dashed lines in FIGS. **1**, **2**A, **4**, and **5**). The embodiment shown in FIG. **2**A includes an additional optional compression module **603** for compressing the unmarked original content stream O_i, which is then provided to XOR operators **110**, **200**, and **202**. When the techniques described in connection with the example embodiment of FIG. **6** are applied to the example embodiments of FIG. **1** and FIG. **2**A that employ a stream cipher, it should be noted that the embedded watermark bit boundaries must coincide with, or be completely within, the compression block boundaries in order to guarantee the presence of full-length watermark bits in the final watermarked content. This requirement can be met by proper selection of watermarking parameters. Besides, most well designed watermarking systems can tolerate a certain level of watermark bit errors. Thus occasional failures to meet the above design criteria should not significantly interfere with proper detection of inserted watermarks. There are also additional watermarking techniques that can further alleviate this problem. These techniques will be described below in the context of block ciphers.

[0081] In a block cipher environment, the operations of the forensic embedder may need further adjustments since two different block types are present: compression blocks and encryption blocks. The former often comprises a variable number of bits white the latter typically comprises a fixed number of bits. For example, in DES encryption algorithm, each block always comprises 64 bits of encrypted data, while a GOP block of compressed MPEG image data may contain any number of bits, depending on the particular nature of the motion picture frame sequence and the compression parameters. In such cases, it may not be possible to align the embedded bit boundaries, the compression block boundaries, and the encryption block boundaries all at the same time. In such cases, the general embedding technique may be modified in accordance with the technique described in U.S. Pat. No. 6,430,301 that was described earlier. Using this technique, embedded logical values may be separated by regions of the original content that are either A) not embedded at all, or B) are embedded with a common watermark in both versions of the embedded content. FIG. **7** shows an example embodiment of this technique used in conjunction with the present invention. Two versions of the original content **701** and **702** are shown to be embedded with logical 1 and logical 0 values, as well as containing Matching Signal Intervals (MSI) blocks **706**, which are identically produced in both versions of the embedded signals **701** and **702**. Each version of the content may be compressed and encrypted. As shown in FIG. **7**, compression blocks **703** may have different lengths and may span several encryption blocks **704**. In practical situations, each encryption block **704** is typically made up of 64 or 128 bits while a typical PEG image frame may comprise several thousands of bits. The simplified diagram of FIG. **7** can be used to illustrate a scenario where there is at least one cutting-and-splicing opportunity within each MSI block **706**. In FIG. **7**, the duration of MSI segments **706** are shown to be equal, but it is understood that MSI segments **706** may be selected to have different lengths. If the watermarking algorithm of FIG. **7** is used for the embedding of the content, the insertion of watermarks may be carried out similar to the technique described in connection with FIG. **4**. The two versions of the encrypted content may be transmitted, on the client side, combined together along the potential splicing points **705** to form an embedded content. The splicing must occur at the encryption block boundaries and must take place within the MSI region. Other variations of the embedding technique, i.e., block-based operations analogous to the ones shown in FIG. **2**A and FIG. **2**B, may be implemented using the embedding algorithm of FIG. **7**. The modifications necessary to produce these systems involve obvious rearrangements of the above-described techniques and will not be discussed further.

[0082] Three design guidelines related to the selection of MSI blocks warrant further attention: 1) there must exist at least one splicing opportunity (splicing point **705**) within each MSI block **706**; 2) MSI boundaries must be aligned with compression block boundaries; and 3) a watermark bit must fit within a single compression block **703**. The first design guideline can be guaranteed by creating MSI segments that are at least twice the size of encryption blocks **704**. The second guideline requires each MSI to span one or more complete compression blocks **703**. Both of these conditions can be easily satisfied since an MSI region may be selected to be the unmarked original content of an arbitrary length. As for the third guideline, in practical situations, each compression block **703** usually comprises several thousands of bits whereas a typical watermark bit only spans a few tens or hundreds of bits. Furthermore, in an unlikely case where a watermark bit can not fit in a single compression block **703**, signal cutting and splicing can occur, for example, at every two compression block boundaries. It is however more likely that each watermark bit is embedded in only a portion of each compression block **703**. This can occur, for example, in an MPEG-compressed signal where a watermark bit is embedded in an I-frame only, while the compression block **703** is a GOP, comprising several additional P and B frames. In such cases, the unmarked portions of the compression block **703** may simply be considered as extensions of the adjacent MSI regions **706**.

[0083] Since the cutting and splicing of the two compressed and encrypted data streams must occur at the compression block boundaries, the boundary locations must be known at the client side. In some cases, it may be possible to preserve the format of the compressed data stream subsequent to encryption. For example, in an MPEG compressed video, only the data within each GOP may be encrypted while keeping some meta data and header information in unencrypted format. This way, while the actual image/audio data is encrypted, the compression block boundaries remain easily recognizable. Such partial encryption of the signal content, however, may weaken the security of the system. In systems where the entire signal content is encrypted, additional synchronization and compression block boundary information may need to be delivered to the client. This can be accomplished by transmitting this additional information together with, or separately from, the embedded content signals. Table 1 below shows an example of how such information may be generated for proper identification of compression blocks.

TABLE 1

Compression	Stream 1 Index		Stream 2 Index	
Block Number	Start Bit	End Bit	Start Bit	End Bit
1	100	5094	100	5090
2 (MSI)	5095	14258	5091	14300
3	15001	15268	14301	15250
4 (MSI)	15269	19269	15251	19298
5	19270	23684	19299	23701
6 (MSI)	23685	35248	23702	35221
. . .	. . .	. . .	. . .	. . .

[0084] For example, according to Table 1, the first compression block starts at bit location **100** in both encrypted streams and ends at bit locations **5094** and **5090** in streams **1** and **2**, respectively. Using the example technique of Table 1, non-data segments of the compressed data streams may be easily identified and avoided when the cut-and-splice watermarking is carried out.

[0085] Table 1 provides only an example embodiment of the present invention and it should be appreciated that there are many different ways of conveying the compression block boundaries. For example, boundary locations may be expressed in terms of encryption block numbers (instead of bit numbers) or they may be expressed in terms relative to other compression blocks. It is also possible to separately identify non-data fields (e.g., headers, metadata, etc.) or other fields of interest within a table similar to the one shown in Table 1. Furthermore, the above described methods may be modified to be used in conjunction with other watermarking techniques that insert digital watermarks into a compressed data stream. One such technique is described in U.S. Pat. No. 5,687,191.

[0086] While specific examples were used in the foregoing disclosure to illustrate the embodiments of the present invention in association with one or more particular configurations of a watermarking system, it is understood that these techniques can be easily adapted to conform to alternate configurations of these watermarking systems. For example, in various embodiments of the present invention, such as those described in FIGS. **1-2** and **4-7**, multiple watermark embedding or encryption modules were presented to facilitate the understanding of disclosed concepts. However, it is understood that a single watermarking or encryption module may perform the necessary operations for ail signal paths. Furthermore, the techniques disclosed in accordance with the present invention can be used in conjunction with data scrambling techniques that may not technically be classified as encryption algorithms. All necessary modifications required to adapt the present invention to such systems are considered to be well within the capabilities of a person of ordinary skills in the art and are not disclosed further.

[0087] It should now be appreciated that the present invention provides advantageous methods and apparatus for watermarking encrypted data streams.

[0088] Although the invention has been described in connection with various illustrated embodiments, numerous modifications and adaptations may be made thereto without departing from the spirit and scope of the invention as set forth in the claims.

1-4. (canceled)

5. A method for facilitating watermarking of an encrypted host content, the method comprising:

producing the encrypted host content at least in-part by:

(a1) obtaining an unencrypted version of the host content,

(b1) embedding a first watermark symbol value into a first set of segments of the unencrypted version of the host content to produce a first embedded host content, and

(c1) encrypting the first embedded host content to produce the encrypted host content;

producing an unencrypted auxiliary information at least in-part by:

(a2) embedding a second watermark symbol value into the first set of segments of the unencrypted version of the host content to produce a second embedded host content, and

(b2) modifying the second embedded host content using the first embedded host content to produce the unencrypted auxiliary information; and

transmitting the encrypted host content and the unencrypted auxiliary information to a remote device to allow insertion of a particular watermark that comprises both the first and the second symbol values in the encrypted host content, the insertion comprising processing a subset of the first set of segments of the encrypted content using corresponding segments from the unencrypted auxiliary information to convert the first embedded watermark symbol value to the second watermark symbol value in each of the subset of the first set of segments without decrypting the subset of the first set of segments.

6. The method of claim **5**, wherein modifying the second embedded host content using the first embedded host content comprises conducting an exclusive OR operation between the first embedded host content and the second embedded host content.

7. The method of claim **5**, wherein processing a subset of the first set of segments of the encrypted content comprises conducting an exclusive OR operation between the he first set of segments of the encrypted content and the corresponding segments from the unencrypted auxiliary information.

8. The method of claim **5**, wherein:

producing the encrypted host content and the unencrypted auxiliary information is carried out at a pre-processing center; and

transmitting the encrypted host content and the unencrypted auxiliary information is triggered by a request for the host content.

9. The method of claim **5**, further comprising, prior to the transmitting, encrypting the unencrypted auxiliary information to produce an encrypted auxiliary content, wherein the encrypted auxiliary content is decrypted prior to processing of the subset of the first set segments of the encrypted content.

10. The method of claim **5**, further comprising:

prior to encrypting the first embedded host content, compressing the first embedded host content;

prior to modifying the second embedded host content, compressing the second embedded host content; and

modifying the second embedded host content using the first embedded host content comprises modifying compressed blocks of the second embedded host content using compressed blocks of the first embedded host content; wherein

processing the subset of the first set of segments of the encrypted host content comprises processing the subset of the first set of segments of the encrypted host content on a compressed-block by compressed-block basis.

11. The method of claim **10**, wherein processing the subset of the first set of segments is carried out on a compressed-block by compressed-block basis without decompressing of the compressed blocks.

12. The method of claim **10**, further comprising embedding a common watermark value into a second set of segments of the unencrypted version of the host content such that the first embedded host content and the second embedded host content include the common watermark value in the second set of segments of the first embedded host content that are aligned with the second set of segments in the second embedded host content.

13. The method for watermarking an encrypted host content, the method comprising:

obtaining the encrypted host content that includes a first watermark value embedded therein, the encrypted host content having been produced at least in-part by:

embedding the first watermark symbol value into a first set of segments of an unencrypted version of the host content to produce a first embedded host content, and

encrypting the first embedded host content to produce the encrypted host content;

obtaining an unencrypted auxiliary information, the auxiliary unencrypted content having been produced at least in-part by:

embedding a second watermark symbol value into the first set of segments of the unencrypted version of the host content to produce a second embedded host content, and

modifying the second embedded host content using the first embedded host content to produce the unencrypted auxiliary information; and

embedding a particular watermark that comprises both the first and the second symbol values in the encrypted host content by processing a subset of the first set of segments of the encrypted content using corresponding segments from the unencrypted auxiliary information to convert the first embedded watermark symbol value to the second watermark symbol value in each of the subset of the first set of segments without decrypting the subset of the first set of segments.

14. The method of claim **13**, wherein modifying the second embedded host content using the first embedded host content comprises conducting an exclusive OR operation between content symbols of the first embedded host content and content symbols of the second embedded host content.

15. The method of claim **13**, wherein processing a subset of the first set of segments of the encrypted content comprises conducting an exclusive OR operation between the he first set of segments of the encrypted content and the corresponding segments from the unencrypted auxiliary information.

16. The method of claim **13**, wherein obtaining the encrypted host content and the unencrypted auxiliary information comprises receiving the encrypted host content and the unencrypted auxiliary information in response to a request for the host content.

17. The method of claim **13**, wherein the encrypted host content and the unencrypted auxiliary information are both in compressed format;

processing the subset of the first set of segments of the encrypted host content comprises processing the subset of the first set of segments of the encrypted content on a compressed-block by compressed-block basis; and

the encrypted host content and the unencrypted auxiliary information having been generated in compressed format at least in-part by:

prior to encrypting the first embedded host content, compressing the first embedded host content,

prior to modifying the second embedded host content, compressing the second embedded host content, and

modifying the second embedded host content using the first embedded host content comprises modifying compressed blocks of the second embedded host content using compressed blocks of the first embedded host content.

18. The method of claim **17**, wherein processing the subset of the first set of segments is carried out on a compressed-block by compressed-block basis without decompressing of the compressed blocks.

19. A device, comprising:

a watermark embedder to obtain an unencrypted version of the host content and embed a first watermark symbol value into a first set of segments of the unencrypted version of the host content to produce a first embedded host content, and to embed a second watermark symbol value into the first set of segments of the unencrypted version of the host content to produce a second embedded host content;

a modification circuit coupled to the watermark embedder to modify the second embedded host content using the first embedded host content to produce the unencrypted auxiliary information;

an encryptor coupled to the watermark embedder and to the circuit to encrypt the first embedded host content to produce the encrypted host content; and

a transmitter coupled to the encryptor and to the modification circuit to transmit the encrypted host content and the unencrypted auxiliary information to another device, wherein insertion of a particular watermark that comprises both the first and the second symbol values in the encrypted host content is enabled at least in-part when a subset of the first set of segments of the encrypted content is processed using corresponding segments from the unencrypted auxiliary information to convert the first embedded watermark symbol value to the second watermark symbol value in each of the subset of the first set of segments without decrypting the subset of the first set of segments.

20. The device of claim **19**, wherein the modification circuit comprises an exclusive OR circuit, one input of the exclusive OR circuit coupled to digital samples of the first embedded host content and another input of the exclusive OR circuit coupled to digital samples of the second embedded host content.

21. The device of claim **19**, further comprising, a second encryptor coupled to the modification circuit and to the transmitter to encrypt the unencrypted auxiliary information to produce an encrypted auxiliary content for transmission to another device.

22. The device of claim **19**, further comprising a compression component coupled to the watermark embedder, to the encryptor and to the modification circuit to compress the first embedded host content and to provide the compressed first embedded host content to the encryptor, and to compress the second embedded host content and to provide the compressed second embedded host content to the modification circuit.

23. The device of claim **19**, wherein the watermark embedder further embeds a common watermark value into a second set of segments of the unencrypted version of the host content such that the first embedded host content and the second embedded host content include the common watermark value in the second set of segments of the first embedded host content that are aligned with the second set of segments in the second embedded host content.

24. A device, comprising

a receiver to receive an encrypted host content and an unencrypted auxiliary information,

the encrypted host content including a first watermark symbol value embedded therein and having been produced at least in-part by embedding the first watermark symbol value into a first set of segments of an unencrypted version of the host content to produce a first embedded host content, and encrypting the first embedded host content to produce the encrypted host content,

the unencrypted auxiliary information having been produced by embedding a second watermark symbol value into the first set of segments of the unencrypted version of the host content to produce a second embedded host content, and modifying the second embedded host content using the first embedded host content to produce the unencrypted auxiliary information; and

a processing circuit coupled to the receiver to process a subset of the first set of segments of the encrypted content using corresponding segments from the unencrypted auxiliary information to convert the first embedded watermark symbol value to the second watermark symbol value in each of the subset of the first set of segments without decrypting the subset of the first set of segments, and to thereby embed a particular watermark that comprises both the first and the second symbol values in the encrypted host content.

* * * * *