

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
2 November 2006 (02.11.2006)

PCT

(10) International Publication Number
WO 2006/116394 A2

(51) International Patent Classification:

G06K 9/00 (2006.01) **G09C 5/00** (2006.01)
H04L 9/00 (2006.01) **H04K 1/00** (2006.01)
G06K 9/36 (2006.01)

(21) International Application Number:

PCT/US2006/015615

(22) International Filing Date: 25 April 2006 (25.04.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/675,231	26 April 2005 (26.04.2005)	US
11/115,990	26 April 2005 (26.04.2005)	US
11/116,137	26 April 2005 (26.04.2005)	US
11/410,961	24 April 2006 (24.04.2006)	US

(71) Applicant (for all designated States except US): **VER-
ANCE CORPORATION** [US/US]; 4435 Eastgate Mall,
Suite 350, San Diego, California 92121 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **PETROVIC,
Rade** [US/US]; 11416 Cypress Woods Drive, San Diego,
California 92131 (US). **TEHRANCHI, Babak** [IR/US];
12570 Spindletop Road, San Diego, California 92129
(US). **WINOGRAD, Joseph, M.** [US/US]; 5378 Adams
Avenue, San Diego, California 92115 (US).

(74) Agents: **LIPSITZ, Barry, R.** et al.; LIPSITZ & MCAL-
LISTER, LLC, 755 Main Street, Building No. 8, Monroe,
Connecticut 06468 (US).

(81) Designated States (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN,
CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI,
GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE,
KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV,
LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI,
NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG,
SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US,
UZ, VC, VN, YU, ZA, ZM, ZW.

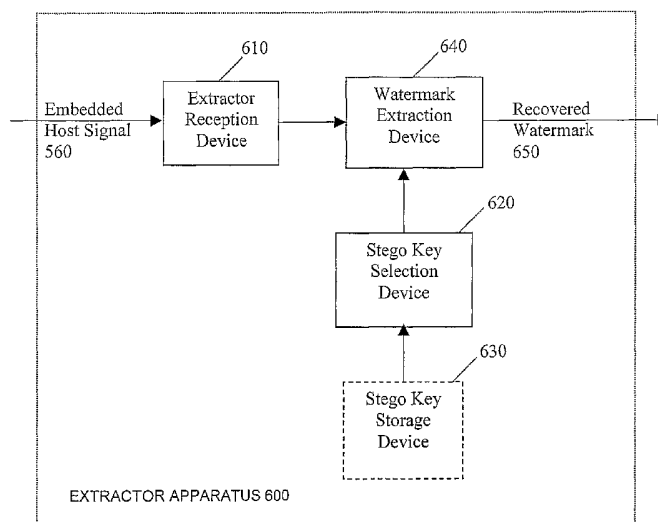
(84) Designated States (*unless otherwise indicated, for every
kind of regional protection available*): ARIPO (BW, GH,
GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,
ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,
RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA,
GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished
upon receipt of that report*

[Continued on next page]

(54) Title: SYSTEM REACTIONS TO THE DETECTION OF EMBEDDED WATERMARKS IN A DIGITAL HOST CONTENT



(57) Abstract: Methods for adapting the operation of a system in response to the detection of embedded watermarks from a digital host content are provided. A digital host content is received and examined for the presence of watermarks. In response to the detection of embedded watermarks and in accordance with the value, type, density or spacing of the detected watermarks, one or more system reactions may take place. These reactions include conditionally allowing the system to resume its normal operation, prohibiting the system from resuming its normal operation, degrading the quality of the digital host content or changing the security status of the digital host content. In response to the extraction of weak watermarks that do not meet the desired system requirements, the extraction operation may be modified or extended to enable the detection of strong watermarks.

WO 2006/116394 A2



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM REACTIONS TO THE DETECTION OF EMBEDDED WATERMARKS IN A DIGITAL HOST CONTENT

This application claims the benefit of U.S. Provisional Application No. 60/675,231, filed April 26, 2005, and is a continuation-in-part of commonly owned, co-pending U.S. Application No. 11/115,990 filed April 26, 2005 and a continuation-in-part of commonly owned, co-pending U.S. Application No. 11/116,137 filed April 26, 2005.

BACKGROUND OF THE INVENTION

The present invention relates generally to systems and methods for insertion and subsequent extraction of digital watermarks from multi-media content. More specifically, the invention relates to providing enhanced security to embedded watermarks in multi-media content such as audio, audiovisual and image content.

Digital watermarks are substantially imperceptible signals embedded into a host signal. The host signal may be any one of audio, still image, video or any other signal that may be stored on a physical medium, transmitted or broadcast from one point to another or received and exhibited using a variety of display means such as monitors, movie screens, audio speakers or print medium. Digital watermarks are designed to carry auxiliary information without substantially affecting the fidelity of the host signal, or without interfering with normal usage of the host signal. For this reason, digital watermarks are sometimes used to carry out covert communications, where the emphasis is on hiding the very presence of the hidden signals. The main applications of digital watermarks include prevention of unauthorized usage (i.e., duplication, playing and dissemination) of copyrighted multi-media content, proof of ownership, authentication, tampering detection, broadcast monitoring, transaction tracking, audience measurement and triggering of secondary activities such as interacting with software programs or hardware components.

The above list of applications is not intended to be exhaustive as many other present and future systems can benefit from co-channel transmission of main and auxiliary information. An example of such a system is one that utilizes a digital watermarks to carry auxiliary informational signals; these signals may convey spatial coordinates (e.g., GPS coordinates) of an apparatus, or timestamps indicating the exact time of generation and/or

transmission of the composite host and watermark signals or any other information related or unrelated to the host signal. Alternatively, digital watermarks may carry information about the content, such as caption text, full title, artist name, and instructions on how to purchase the content. Other applications of watermarks include document security and counterfeit prevention for printed materials. In such applications, the presence of hard to re-produce (e.g., hard to copy) watermarks establishes authenticity of the printed material.

There is a considerable amount of prior art describing various digital watermarking techniques, systems and applications. Watermarking techniques described in the literature include methods of manipulating the least significant bits of the host signal in time or frequency domains, insertion of watermarks with an independent carrier signal using spread spectrum, phase, amplitude or frequency modulation techniques, and insertion of watermarks using a host-dependent carrier signal such as feature modulation and informed-embedding techniques. Most embedding techniques utilize psycho-visual or psycho-acoustical analysis (or both) of the host signal to determine optimal locations and amplitudes for the insertion of digital watermarks. This analysis typically identifies the degree to which the host signal can hide or mask the embedded watermarks as perceived by humans.

In most digital watermarking applications, the embedded watermarks must be able to maintain their integrity under various noise and distortion conditions that may affect the multimedia content. These impairments may be due to various signal processing operations that are typically performed on multimedia content such as lossy compression, scaling, rotation, analog-to-digital conversion, etc., or may be due to noise and distortion sources inherently present in the transmission and/or storage channel of multi-media content. Examples of this type of noise include errors due to scratches and fingerprints that contaminate data on optical media, noise in over-the-air broadcasts of audio-visual content, tape noise in VHS tapes, everyday handling of currency notes, and the like. Typically, increased robustness of embedded watermarks may be obtained at the expense of reduced transparency of the watermark.

The security of digital watermarks is another aspect of watermarking systems. In certain applications such as proof of ownership, source authentication, piracy tracing, access control of copyrighted content, and the like, it is essential that embedded watermarks resist intentional manipulations aimed at detecting the presence of watermarks, deciphering the data

carried by the watermarks, modifying or inserting illegal values (forgery), and/or removing the embedded watermarks. To this end, many watermarking systems employ a secret key to enable embedding and subsequent extraction of the watermarks. These systems should be distinguished from cryptographic systems where a secret key is used to prevent unauthorized access and/or modification of the information but are not designed to prevent the detection of the presence or the removal of the encrypted information. Such cryptographic systems, depending on the length of the key and the complexity involved in breaking the key, could theoretically guarantee security of encrypted digital data for most practical situations. Indeed, cryptography can be used to protect against unauthorized reading or forgery of watermark data, but it fails to provide protection against other types of attacks that are aimed at preventing the legitimate users from detecting or extracting the embedded watermarks altogether. By the way of example and not by limitation, these attacks include synchronization attacks, replacements attacks and noise attacks that modify the composite host and watermark signal in such a way to obscure or damage the embedded watermarks beyond recognition. More details on possible attacks will be presented below.

Designing a watermarking system requires reaching the proper balance between transparency (imperceptibility), robustness and security requirements of the system. A fourth requirement is the watermark payload capacity. This requirement depends on the specific application of the watermarking system. Typical applications range from requiring the detection of only the presence of watermark (i.e., single-state watermark) to requiring a few tens of bits of auxiliary information per second. In the latter case, the embedded bits may be used to carry identification and timing information such as serial numbers and timestamps and metadata such as captions, artists names, purchasing information, and the like.

A fifth factor in designing practical watermarking systems is computational costs of the embedding and/or extraction units. This factor becomes increasingly important for consumer electronic devices or software utilities with limited silicon real estate or computational capabilities. This factor is strongly related to the application at hand. For example, watermarks for forensic tracing of piracy channels, such as those that embed different codes in each copy of content distributed over the Internet, may require a simple embedder but a complex and costly forensic extractor. On the other hand, copy control systems designed to prevent

unauthorized access to multimedia content, for example, in consumer electronic devices, may tolerate a sophisticated embedder but require a simple and efficient extractor.

The sixth important factor in designing a practical watermarking system is the probability of false detections. Again, this requirement varies depending on the application at hand. In certain applications, such as copy control, the probability of false detections must be very low since executing a restrictive action on a legally purchased content is bound to frustrate users and have negative implications for device manufacturers and/or content providers. On the other hand, in broadcast monitoring systems where the frequency of broadcast content is measured to generate royalty payments or popularity charts, much higher false detection rates may be tolerated since the presence of a few false detections may have very little effect on the final outcome of the counts.

The prior art systems, at best, use an ad-hoc approach for designing watermarking systems that happen to have certain collection of features, which are then mapped onto various applications in search of a good match. These systems also fail to systematically analyze security threats and provide answers to different threat scenarios. For example, U.S. patent No. 5,889,868 (Moskowitz, et. al.) discusses randomizing the insertion locations of watermarks within the content signal as well as varying the embedding algorithm throughout the content. But there are no enabling embodiments that describe how this randomization may take place and how this would affect a watermarking system's design parameters. This reference also merely states that at any given location of a content one or another embedding technique may be used but it fails to discuss simultaneous utilization of embedding technologies. It also fails to discuss joint configuration of embedders and extractors in order to vary levels of robustness/security/opacity/cost. In another prior art system as disclosed by D. Kirovski, et. al., in "Multimedia Content Screening Using a Dual Watermarking and Fingerprinting System", Tech. Rep. MSR-TR-2001-57, Microsoft Research (June 2001) a technique is employed in which the host content is embedded in a conventional way (e.g., using a spread spectrum technique) using a secret watermarking key (SWK). The detection key for each detector, however, is different from SWK. The individualized detection key is generated by adding noise to SWK. Since detection is done via correlation, the noise-contaminated detection key should still produce the desired correlation value if there are no other significant (additional) impairments present. To build up immunity against additional impairments and

more aggressive attacks, the length of the spreading sequence may be increased to compensate for the robustness penalty incurred due to non-optimum detection key. The techniques discussed in this prior art, however, are different from the present invention in many ways. First, the embedding is done in a conventional way so the variations in embedding space as well as the relative size of embedding space to the detection space are not considered. Second, detection keys constitute a degraded version of the embedder key; this produces a degraded correlation value during the detection process. In the present invention, however, individual detection keys are not generated by adding noise to the embedder key and the correlation value in the detection process is not degraded. Further, this reference also fails to discuss how the robustness/security/transparency needs of the watermarking system can be addressed using a systematic design approach that is suitable for a multitude of applications and needs.

These and other shortcomings of the prior art systems are addressed by the methods and apparatus of the present invention.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide methods for design and implementation of digital watermarking systems that overcome various deficiencies of the prior art systems. It is another object of the present invention to provide systematic methods for designing watermarking systems that are arbitrarily secure, robust, transparent to the user, reliable, and cost effective. It is also an object of this invention to anticipate circumvention attempts against the disclosed watermarking systems and render such attempts ineffective. It is another object of the present invention to adapt the operation of systems in response to watermarks detected in or extracted from a digital host content. Another object of the present invention is to resolve uncertainties in the detection of embedded watermarks in a digital host content that may arise due to the detection of multiple watermark states, inadvertent capture of watermarks, or the presence of content with multiple credentials. It is also an object of the present invention to increase the payload capacity of an existing watermark system without sacrificing robustness, computational complexity or security of the system. It is a further object of the present invention to reduce the computational complexity of a multi-channel watermark extraction system.

These and other objects of the present invention may be accomplished, for example, by subjecting a received digital host content to watermark detection and /or extraction. If none of the detected or extracted watermarks comprise a strong watermark, but at least one weak watermark is detected, the extractor operation may be modified. This modification may comprise selecting a different extraction stego key, or may comprise extending the extractor operation to search at least one additional pixel area, time segment, spatial or temporal frequency range, color component, audio channel, or one or more components of a compressed version of the digital host content. In addition, such modification may comprise increasing the granularity of watermark extraction, employing additional error correction code techniques, or undertaking a forensic analysis of the digital host content.

In response to the detection of watermarks from a digital host content, the system may be permitted to continue its normal operation, may be conditionally permitted to resume its operation, or may be prohibited from its normal operation. Furthermore, the status of the digital host content may be modified or its quality may be degraded.

A conditional permission may comprise receiving a response from a user, a database, or a third party. A prohibitive action may comprise muting the audio or blanking the video portions of the digital host content, or stopping the playback, recording or transfer of the digital host content.

Further reactions may comprise modification of the embedded watermarks or the credentials associated with the digital host content. Additionally or alternatively, the digital host content may be fully or partially obscured. Such obscurations may or may not be reversible.

Some content degradation techniques may comprise down-sampling, lossy compression, dynamic range reduction, partial scrambling, spectral shaping, addition of wow and flutter, addition of noise, phase distortion, or intermittent blanking or muting of the digital host content. While the digital host content may be restored to its original form if such degradation techniques are reversible, non-reversible techniques may permanently degrade the quality of the digital host content.

The system operation may further be adapted so that in response to the detection of multiple watermark states, a distinct enforcement rule is associated with each watermark state. The most restrictive enforcement rule associated with the extracted watermarks may be selected. Alternatively, it may be possible to select the least restrictive enforcement action associated with the extracted watermarks, or a restrictive enforcement action somewhere between the most and least restrictive enforcement action.

The system operation may also be adapted so that in response to the detection of multiple watermarks from a digital host content from at least two monitoring intervals, an enforcement action is commenced when at least a first watermark state is detected from two or more monitoring intervals. The enforcement action may further commence if the watermark state is detected in accordance with a predefined value, type, density or spacing. The enforcement action may comprise at least one of permitting the normal operation of the system, conditionally permitting the operation of the system, prohibiting the normal operation of the system, modifying the status of the digital host content, or degrading the quality of the digital host content. This enforcement action may last for a pre-determined period of time or may vary randomly. Furthermore, the detection of each additional watermark state may further extend the enforcement action.

In order to reduce the computational complexity of watermark extraction from a received multi-channel digital host content, a subset of the received channels may be selected and combined to form a composite signal that is subjected to watermark extraction. The selection of the subset of received channels may be done in accordance with a probability value. This probability value may have a uniform or non-uniform distribution. Furthermore, this probability value may be calculated in accordance with the number of channels of the multi-channel digital host content. Once a particular channel combination is selected, watermark extraction may persist for a period of time, at the end of which, a new combination of channels may be selected for further watermark extraction. The duration of this period of time may be random.

Furthermore, the combining of the subset of channels may comprise obtaining a coefficient associated with each selected channel and adding together the selected channels in accordance with the associated coefficients. These coefficients may all be of equal value or may be selected in accordance with the characteristics of the selected channels. Additionally, they may be selected in accordance with a probability value.

Other features provided by the present invention include expanding the payload of an existing watermarking system by allocating a first set of embedding opportunities to a first set of payload values, and at least a second set of embedding opportunities to a second set of payload values. This way, a single tier watermarking system is converted into a multi-tier watermarking system, where each tier comprises a particular set of embedding opportunities, and each tier comprises a set of payload values that is, at most, as large as the original single tier (i.e., the base tier). While each watermarking tier by itself may not exceed the payload capacity of the original system, when two or more tiers are combined together, an increase in payload capacity is realized.

BRIEF DESCRIPTION OF THE DRAWINGS

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

Fig. 1 illustrates an example embodiment of a self synchronizing scrambler;

Fig. 2 illustrates an example embodiment of a stego key;

Fig. 3 is a table describing an example of various constituents of an embedding stego key;

Fig. 4 is a block diagram showing an example embodiment of an Embedding Apparatus in accordance with the present invention;

Fig. 5a is an example pictorial representation of an embedding stego key set and an extraction stego key set in the absence of distortions;

Fig. 5b is an example pictorial representation of an embedding stego key set and an extraction stego key set in the presence of distortions;

Fig. 6a is an example pictorial representation of multiple embedding stego key sets;

Fig. 6b is an example pictorial representation of multiple extracting stego key sets;

Fig. 6c is an example pictorial representation of multiple embedding and extraction stego key sets;

Fig. 7 is an example of a table describing the relationships between the system characteristics and stego keys;

Fig. 8 is a block diagram showing an example embodiment of an Extractor Apparatus in accordance with the present invention;

Fig. 9 is an example of a time-frequency diagram used for illustrating the concepts described in Example 1;

Fig. 10 is an example of a table describing embedder stego key distribution related to Example 1;

Fig. 11 is an example of a table describing extractor stego key distribution related to Example 1;

Fig. 12 is an example of a table describing the effects of extractor circumvention related to Example 1;

Fig. 13 is a first example graph describing the relationship between various system characteristics;

Fig. 14 is a second example graph describing the relationship between various system characteristics;

Fig. 15 is a third example graph describing the relationship between various system characteristics;

Fig. 16 is a fourth example graph describing the relationship between various system characteristics;

Fig. 17 is a fifth example graph describing the relationship between various system characteristics;

Fig. 18 is an example of a table describing an example of soft decision parameters;

Fig. 19 is a flowchart illustrating a masking technique in accordance with an example embodiment of the invention;

Fig. 20 is a flowchart illustrating a masking technique in accordance with a further example embodiment of the invention;

Fig. 21 is a flowchart illustrating a masking technique in accordance with a further example embodiment of the invention;

Fig. 22 is a flowchart illustrating an embedding prevention technique in accordance with an example embodiment of the invention; and

Fig. 23 is an example of a table describing watermark payload expansion in accordance with an example embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

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. Further, the concepts that are described herein are generally and specifically applicable to all systems comprising methods and apparatus for embedding and/or extracting digital watermarks.

Throughout the description of the various embodiments of the present invention, the words “extraction” and “detection” are sometimes used interchangeably in reference to the discerning of the presence and/or value of the embedded watermarks from a digital host content. However, it should be appreciated that, in the context of the present invention, watermarks may be detected in the digital host content without being extracted therefrom. While the detection and/or extraction of watermarks may lead to the recovery of complete and identifiable watermarks, it may also lead to the recovery of incomplete, weak or otherwise unreliable watermarks that require further processing and/or analysis.

The invention described herein provides guidelines for the design of watermarking systems that can be easily adapted for various applications. The transparency (a.k.a. imperceptibility, or fidelity) of embedded watermarks may be a critical design element for many multimedia applications. This is especially true for certain applications where significant creative and financial resources have been utilized to produce just the right audio-visual experience. In such cases, any noticeable artifacts due to the insertion of watermarks may alter the intended effects of the audio-visual content and is simply unacceptable. For other applications, however, the transparency requirements may be somewhat less stringent. For example, prior to reaching the intended audience in a typical consumer electronic environment, the multimedia content may undergo several signal processing operations, such as compression, A/D and D/A conversions, equalization, broadcasting, etc., that introduce processing artifacts into the multimedia content. In such cases, it may suffice for the

watermark-induced artifacts not to exceed the ones produced by such signal processing operations.

Watermark robustness is also a very important requirement for multimedia applications. Generally, watermark robustness refers to immunity against standard processing of the host signal, such as perceptual data compression, A/D and D/A conversions, resampling, standard content editing, etc. Robustness is also closely related to the restrictions imposed on probability of false detections. False detections may occur in two different ways. First, for any watermark extraction system, there is a small, but finite, probability of falsely detecting a watermark in an unwatermarked content. SDMI and DVD-audio requirements specify a false positive probability of less than 10^{-12} per 15-second detection interval. This means that the average time between false positives is 476 thousand years of continuous monitoring of unwatermarked content. A second type of false detection occurs when a watermarked content is being examined and a particular watermark value is falsely recognized as a different watermark value. This type of false detection, normally called a mis-detection, can produce unwanted outcomes. For a properly designed watermarking system, the rate of mis-detections should be roughly the same order of magnitude as the rate of false detections described above.

Another important factor in designing a watermarking system is the computational complexity of watermark extractors. This requirement can be stated as maximum Millions of Instructions Per Second (MIPS) value, maximum gate count, maximum ROM and RAM size, etc. In principle, the watermark extractor cost should be a small fraction of the cost of the device, or its processing load should amount to a small fraction of the processing load of the host software module.

The most challenging aspect of a proper watermarking system design involves the fulfillment of security requirements. Designing such systems may be particularly difficult for multimedia content applications since a watermark embedder and/or extractor may be readily available to (or may be acquired by) an attacker of the system. The attacker may then discover circumvention techniques that include altering, disabling or eliminating all or some of the embedded watermarks. These circumvention techniques may be published, distributed or even implemented as "black box" hardware or software modules that are sold to the general public. It is further possible for this breach to become permanent since it may be very difficult to provide security upgrades (e.g., improved watermarking embedding or detection algorithms) to

devices that have already been deployed. Another limitation that further constrains the design of these systems is that it is often required to implement inexpensive embedding and/or extraction designs that fit into environments with limited computational capabilities.

Watermark Technologies

Two classes of watermark algorithms that have received the most attention in the scientific community are Quantization Index Modulation (QIM) and spread spectrum (SS) based watermarks. In Spread Spectrum systems, the watermark data (bits) are modulated by a “chip sequence” in the form of a pseudo random string of +1 and –1 values. On the extractor side, the received signal is multiplied with the same pseudo random string, in sync with the sequence used in embedder, followed by an “integrate-and-dump” extraction of watermark bits. Typically, the extraction process consists of two steps. In the first step, called sync acquisition, the extractor attempts to match its local chip sequence generator with the embedded chip sequence. Once this is successfully achieved, the extractor switches to a data collection and sync maintenance regime.

The need to achieve synchronization between the embedded chip sequence and the locally generated chip sequence presents a major challenge for the Spread Spectrum watermarks. Many attacks are designed specifically to break down this synchronization; for example, by varying the playback speed of a multimedia signal, deleting or repeating randomly selected rows and columns in a picture, etc. An alternative technology, which overcomes the synchronization issues of Spread Spectrum systems, utilizes a “replica signal” derived from the content itself, in place of the pseudo random string. The replica signal is used to modulate the watermark data prior to embedding, and again, at the receiver, the replica signal is generated and used for demodulation, followed by an “integrate-and-dump” extraction of the embedded data. Any time shifts of audio signal or geometric shifts in images would simultaneously affect the replica and the watermark signals, and their synchronization would be substantially maintained.

Attacks on Watermarks

The security goals of a watermarking system may include providing immunity against forgery, eavesdropping, erasure and modification of watermarks. Proper design of a watermarking system, in accordance with the embodiments of the present invention, includes

security features that are designed to thwart all of the above circumvention techniques. These features provide for steganographic embedding of imperceptible watermarks that are difficult to detect, analyze, forge, erase or modify. Thus, the embedded watermarks are immune to a variety of attacks that may be undertaken by an attacker. Some of these attacks will be further described below. Additional techniques, for example, encryption, hashing, generation of digital signatures and other non-watermarking techniques may also be incorporated into the present system to provide additional levels of protection.

In order to design a secure watermarking system and predict its effectiveness, different attack methodologies on watermarking systems must be studied and countermeasures must be developed. The degree of success for each class of attack depends on the particular watermarking technology, particular multimedia content type and the intended use of the multimedia content. While some watermarking algorithms may be inherently immune to one type of attack, they may perform poorly against a different class of attacks. In addition, the resiliency of watermarks that are embedded based on psycho-visual and psycho-acoustical principles depends on the properties of the host content. Thus, it may prove to be more difficult to remove the same watermark from one content than from another. Finally, the usage of the content may set the bar on success or failure of an attack. For example, viewing of a deteriorated multimedia content with no watermarks may be tolerated on a small hand-held device in a noisy setting, such as an airport, but the viewing of the same content may not be acceptable on a home theatre HDTV screen.

Blind Attacks

In the case of a blind attack, an attacker does not have, and does not try to obtain, any information about the watermarking technology and its secret parameters (a.k.a. stego key). The attack is performed by applying different signal distortions to the content and finding one that removes the watermark with the least amount of damage to the host signal. Many blind attacks appear in the form of “benchmark” software packages like Stirmark, Certimark, Checkmark, etc. Detailed description of these test benchmarks can be found in many published documents.

Some blind attacks are more sophisticated than the simple application of various distortions. For example, the blind pattern matching attack, described in Kirovski et al., is based on automatic detection and swapping of similar fragments in watermarked content. It is

argued that most multimedia content is comprised of repetitive segments that can be swapped with each other. This action would disrupt the continuity of embedded watermarks but would maintain an acceptable perceptual quality due to the similarity of swapped segments. This underlying assumption requires the presence of similar segments throughout the content. While this may be true for a limited subset of multimedia content, this type of attack is not likely to be effective in every type of content. It is anticipated that in some types of content it would be unlikely to find enough similar segments for swapping while maintaining a reasonable audio-visual quality and at the same time, removing all embedded watermarks.

Informed Attacks

It is generally assumed that some knowledge of the watermark embedding and extraction methodology is beneficial in devising more efficient attacks for circumventing the system. In other words, an attacker with partial knowledge of the watermarking algorithms and parameters may be able to devise a successful attack that produces a smaller amount of perceptible distortion compared to its blind counterpart. The simplest form of informed attack is to determine the source of watermarks within the components of the multimedia content. In such cases, the multimedia content may be separated into audio and video portions and each type may be tested separately or decomposed into further components to determine the source of embedded watermarks. Obviously, this method may not be useful if all multimedia components contain watermarks. Most informed attacks are more sophisticated. These attacks are discussed below.

Differential Analysis and Attack

For some watermarking systems, it may be reasonable to assume that the original, unwatermarked content (in addition to the watermarked version) is available to the attacker. By comparing the two versions, an attacker may discover certain facts about the embedding technology as well as certain secret parameters about the embedded watermarks. This knowledge may be sufficient to undo the watermarking, or to overwrite it with a different message, or simply select an attack that is most damaging against the particular algorithm and its now-discovered parameters.

The procedure for conducting a differential attack includes subtracting the watermarked content from the unwatermarked signal and analyzing the result. It should be noted that other techniques for generating a "difference signal" may examine the ratio of watermarked to unwatermarked signal (or its inverse). It is further possible to first align the two signals in order to compensate for possible signal distortions and then generate the difference signal in some of the ways described above. This alignment may be performed in time domain as the two versions of the multimedia content are manipulated so that salient points (i.e., portions of the multimedia content with distinctive features such as waveform peaks in audio signals, certain color components, scenes and/or frames in video and image signals) are aligned with each other. Obviously, similar alignment procedures and/or difference signal generation can be carried out in temporal or spatial frequency domains. The alignment procedure could require linear or non-linear amplitude, time domain or frequency domain scaling, as well as rotation, cropping and other geometrical manipulations. Once proper alignment is achieved, analysis of the difference signal may yield valuable information regarding embedding techniques and/or its secret parameters.

By the way of example and not by limitation, such analysis may include examination of the frequency range of the difference signal, the amplitude of the difference signal, the energy of the difference signal, the phase of the difference signal, the absolute value of the difference signal, the sign of the difference signal, the periodicity of the difference signal, and any combination of the above in relation to certain attributes or features of the host content. For instance, it may be discovered that when the auto-correlation of the host signal over a certain time segment crosses a certain threshold, some characteristic of the difference signal (e.g., its energy, sign, mean, etc.) changes in a particular way. In general, relationships between the activities in the difference signal and the host content attributes and features can be used to discover other aspects of the watermarking technology.

Alternatively, the original and watermarked signals may be transformed into frequency domain, wavelet domain, or some other 'transformed' domains, and then the difference between transformed signals may be observed. This would be useful, for example, if the watermarking system contains a linear transformation and by analyzing the difference signal in frequency domain it may be discerned whether this transformation includes phase or magnitude changes or both.

It is further possible to determine whether the presence of the host signal is required for extraction of watermarks by trying to extract the watermark from the difference signal or a modified version of the difference signal. Such analysis could bear fruit if a foreign, additive or multiplicative carrier signal is used to carry the watermark information. In basic spread spectrum systems, for example, the embedded watermarks may be extracted from either the composite host and watermark or just from the difference signal. In feature modulation techniques, on the other hand, the presence of host is required for the recovery of the embedded watermarks, i.e. the difference signal only cannot be used to extract the watermarks.

Collusion Attacks

In this case, the attacker may obtain several copies of the same content containing different watermarks. Again, the attacker may attempt to perform differential analysis on a multiplicity of difference signals in order to discover certain details of the watermarking algorithm and/or its secret parameters. On the other hand, in the case where all watermarks convey the same message (e.g., the same ID number), one simple analysis would entail determining whether or not the multiplicity of difference signals are identical or whether or not there is a simple relationship, such as a time shift, between the various difference signals. This may indicate the presence of additional information, such as forensic information, fingerprints, etc., or may indicate that embedding of the individual copies is done separately, with a different set of parameters.

In a different collusion attack scenario, an attacker may obtain several completely different contents containing the same watermark and produce an averaged signal. If the host contents are mutually independent random variables and there is a large number of samples, the result of averaging should be approximately a constant (DC) signal. If watermarks are identical and signal independent, it would be superimposed onto this DC component and ready for further analysis.

In a different scenario where copies of the same content with different watermarks (i.e., different ID numbers) are available, the attacker can employ alternative collusion strategies for removal or obscuration of the embedded watermarks. For example, the attacker can generate a single copy of the content by averaging several copies that contain different watermarks. The averaging process should not appreciably deteriorate the perceptible quality of the host content

but could generate enough “noise” in the watermark signal to interfere with reliable detection of the embedded watermarks. Further enhancements of this technique may require weighted and/or non-linear averaging schemes. Other variations of the collusion attack could involve cutting each copy of the content into small pieces, and then assembling a new copy by concatenating the different pieces together in either a deterministic or random (or pseudo-random) way. For example, an attacker may select a segment with minimum, median or maximum energy, and include this segment in an output copy.

Oracle Attack

In the presence of a watermark extractor that provides only two states – watermark detected or not detected – it is possible to design procedures to estimate the embedded watermarks. This estimate can be used for watermark removal, overwriting, analysis, etc. The procedure consists of bringing the host content just outside the watermark detection region by iteratively adding distortion to the content in small steps, e.g. by adding small amounts of noise. Next the attacker would make small changes in various places and note those changes that switch the watermark detector on again. The collection of those changes represents an estimate of the watermark.

Embedder Based Attacks

It may be a reasonable to expect the attackers to gain access to one or more watermark embedders. In such cases, two types of attacks on the watermarking system may be expected: Overwriting Attack and Analysis attack. Overwriting attack is an attempt to embed a different (and perhaps more permissive) watermark in place of the originally embedded watermark. With this type of attack, there is no need to analyze the watermarking technology or its secret parameters. Effectiveness of this type of attack depends on the particular watermark embedding technology and its implementation. For example, this type of attack may be effective against a spread-spectrum watermarking system that uses the same carrier signal for embedding all watermarks. However, if several different carrier signals are employed at random to effect embedding, the attackers’ attempts at overwriting may introduce additional watermarks without erasing the original. Even if the same PN sequence is used to carry different data bits, overwriting attacks require more or less perfect synchronization between the

original embedding and the overwriting attempt in order to completely obliterate the original watermark. This condition is also true for feature modulation techniques where misalignment of the two watermarking layers could result in the extraction of two distinct watermarks, albeit one layer may be stronger than the other.

Another type of attack that may be launched by having access to an embedding device is a form of denial-of-service attack. An attacker may intentionally embed a second set of watermarks to confuse and/or mislead the watermark extraction unit and inconvenience and frustrate legitimate users of the content. For example, in a copy control application, a more restrictive state, such as "do not copy" may be inserted into a content originally containing "copy freely" watermarks. Upon the detection of both watermark values, a recording device may decide to act in accordance with the usage rules of the more restrictive of the pair and disable the recording of the content. Similarly, in a broadcast monitoring application, a content may be originally embedded with the content owner's name and a serial number. An attacker may attempt to add a new set of names and serial numbers to the content prior to the broadcast of the segment. In this case, the presence of two or more conflicting set of watermarks may result in inaccurate reporting of broadcast monitoring information.

An "Analysis Attack" is similar to the type of attack described in the context of differential attacks, except that in the presence of an embedder, an attacker has the option of using specially designed test signals in order to extract secret parameters of the watermarking technology. Examples of such test signals include, but are not limited to, impulse signals, sinusoidal signals, flat image signals, edge image signals, step functions, signals with specific temporal or frequency characteristics and other specially formulated mathematical functions. An attacker may be able to, for example, determine the transfer function of the watermarking system (applicable for linear systems only) by analyzing the impulse response of the system. Analysis of the embedded signals of finite time duration or limited frequency range could also provide information regarding the minimum duration of watermarks, their frequency range, etc. In short, it may be possible to generate specially designed test signals of arbitrary complexity, embed them with watermarks and analyze them to gain some knowledge about the watermarking system.

Protocol Attacks

This type of attack forgoes attacking the watermark altogether in order to convert a compliant device into a noncompliant device. For example, it may be possible to disable the communication link between a watermark extractor and device control circuits, either by hardware or software modifications. It may be further possible to generate dummy signals (i.e., spoof signals) to deceive a compliant device into providing access to an otherwise restricted content.

Alternatively, it is possible to design a signal modifier that would transform the watermarked content in such a way that the host signal and/or the embedded watermarks are not recognizable by detection devices. The modified content may escape scrutiny by the compliant device since 1) it may not contain a watermark in recognizable form and/or 2) it may not be recognized as an audio-visual signal by the device. Once the modified content is loaded into the compliant device, it may be transformed back to its original form and accessed for further use. It is particularly simple to design a digital scrambler that would perform a random permutation of an input bit stream so that it does not resemble the original signal. By the way of example and not by limitation, the following self-synchronizing scrambler can adequately do this job. The scrambling of input bit stream $b(n)$, $n=1,2,3,\dots$ (n represents the sequential bit number) into output bit stream $c(n)$, $n=1,2,3,\dots$ can be achieved by the following operation:

$$c(n) = b(n) \oplus c(n-1) \oplus c(n-M) \quad \text{Equation (1)}$$

where operator \oplus represents modulo 2 addition, $c(n) = 0$ for $n < 1$ and M is a positive integer greater than or equal to 2. This operation is perfectly reversible and the original bit stream can be recovered using formula:

$$b(n) = c(n) \oplus c(n-1) \oplus c(n-M) \quad \text{Equation (2)}$$

Note that $c(n-1) \oplus c(n-1) = 0$, as well as $c(n-M) \oplus c(n-M) = 0$, while $b(n) \oplus 0 = b(n)$ and commutation and association rules are valid with respect to modulo 2 addition. Fig. 1 illustrates the scrambling and descrambling carried out in accordance with Equations 1 and 2. In the absence of any channel distortions, the signal stream b'_n should be identical to the original bit stream b_n . As noted above, the scrambling algorithm described above provides only an example of many possible signal modification techniques which may be used in this type of

attack. In general, any reversible algorithm that modifies the signal in a way to obscure the presence of embedded watermarks, and/or the host content, may be employed. Furthermore, perfectly reversible signal modifications/transformations may not be required as long as 1) the modified signal escapes detection by compliant devices and 2) the signal retains reasonable perceptual quality subsequent to its transformation back to the original format.

It is also possible to do similar signal modifications in such a way that descrambling can be done in the analog domain. An efficient and effective method is based on multiplication (modulation) of original data with a foreign function. In the case where analog signal is a time function $s(t)$, signal modification can be done by multiplying it with a foreign function $f(t)$, to obtain scrambled signal $c(t)$:

$$c(t) = s(t)f(t) \quad \text{Equation (3)}$$

The descrambling can be done by multiplying the scrambled signal with $f(t)$ followed by Low Pass Filtering (LPF):

$$s'(t) = \text{LPF}(c(t)f(t)) = \text{LPF}(s(t)f(t)^2) \quad \text{Equation (4)}$$

Note that $f(t)^2$ is expected to have a strong DC component, which would multiply $c(t)$ to provide the original signal $s(t)$. Depending on selection of function $f(t)$ there may be some other components that would affect output signal quality. For example, an estimate of the original signal may be obtained by calculating the mean value of $f(t)^2$ according to the following:

$$s'(t) = \text{mean}(f(t)^2)s(t) + n(t) \quad \text{Equation (4)}$$

With proper selection of $f(t)$, the noise $n(t)$ can be minimized. For example, in the case of an audio signal with upper cutoff frequency f_c , multiplication with $f(t) = \cos(2\pi f_c t)$ would invert the spectrum of the input signal, while descrambling (another multiplication with $\cos(2\pi f_c t)$) would restore the original spectrum. The noise signal would appear in frequency domain above f_c , which would be filtered out by a low pass filter, or may be left alone as an insignificant (substantially imperceptible) noise.

Design Principles

It is often possible to redundantly embed many watermarks within a single content. In most applications, a successful attack has to be able to remove the majority of embedded watermarks from a multimedia content. In some applications, a successful attack must eliminate all embedded watermarks since even a single watermark extraction could trigger a restrictive action. The embedded watermarks may be embedded in a variety of domains. For example, if a movie lasts for one-and-a-half hours and a watermark lasts for one second, there will be 5,400 potential time intervals for the insertion and subsequent extraction of the watermarks within this content. Similarly, for an audio signal that spans a frequency spectrum of 16 kHz and a watermark that spans 200 Hz, there will be 80 different frequency bands that can be used for watermark insertion and subsequent extraction. In the case of a video signal, the watermark may be embedded only in a fraction of the screen area. For example, if the screen contains 1024x768 pixels, and the watermark requires an 8x8 pixel area, then there will be $(1024 \times 768) / (8 \times 8) = 12,288$ different places where a watermark can be hidden. If, further, multiple domains are used, e.g. the frequency, space, and time domain watermark embedding, the attacker will have the daunting task of removing watermarks from a very large number of possible hiding places.

Emulation of Cryptographic Systems

Before describing the design principles for a secure watermarking system, it is necessary to review some of the related concepts associated with cryptographic systems. Typically, a successful attack on a crypto-system requires the knowledge of two entities: the encryption algorithm and the encryption key. Assuming that such algorithms are perfectly secure (i.e., there are no back doors), the problem of attacking the system becomes the problem of finding the correct encryption key value. The description of different types of stego-systems is well documented in many scientific texts and articles. Encryption algorithms can generally be categorized into two groups: symmetric systems and asymmetric systems. The difference between the two categories is in the usage and complexity of encryption and decryption operations. In symmetric systems, the same encryption key is used for both the encryption (encoding) and the decryption (decoding) and the two operations generally mirror each other. In asymmetric systems, encryption keys of different lengths are used for encoding and

decoding operations, and typically, the two operations differ in complexity. Regardless of the classification of the crypto-system, launching a successful attack on an encryption system may contain the following four steps:

Step 1: obtain a plaintext (i.e. original content),

Step 2: obtain a ciphertext (i.e., encrypted version of the plaintext),

Step 3: select an encryption key value, and

Step 4: decrypt the ciphertext and compare it to the plaintext.

This is referred to as the known-plaintext attack. There are several other types of cryptanalysis techniques, such as, the ciphertext-only attack, chosen-plaintext attack, adaptive-chosen-plaintext attack, chosen-ciphertext attack, chosen-key attack, etc. The object of all these attacks is to recover the correct encryption key value (or a methodology to recover the encryption key value) in order to decrypt all other messages that are encrypted by the same key. There are other methods for obtaining the encryption key that are listed in the literature, some of which include bribing or coercing individuals to reveal secrets of the encryption system. These types of attacks are very effective and pose a serious threat to the security of cryptographic systems. It is a further object of the present invention to design a watermarking system to greatly reduce the success of attacks that rely on any individual's knowledge of the system.

The most basic attack against the encryption key space is the brute force approach where, based on the knowledge of the encryption algorithm, the attacker repeats steps 1 through 4 above 2^n times in order to guarantee the recovery of an n-bit encryption key (on average, $(2^n)/2$ iterations are sufficient). Thus, the difficulty of a successful brute-force attack grows exponentially with key length. There are certainly more efficient algorithms for attacking the encryption key space but they usually require advanced knowledge of the cryptographic system. Future quantum computing techniques may also render the current cryptographic systems obsolete, altogether. Currently, for a symmetric encryption system with no back doors, a key length of at least 128 bits is considered sufficiently large for withstanding different types of attacks.

In order to further increase the security of crypto-systems, it is possible to divide the content that is being protected into several segments and use a different encryption key and/or algorithm for each segment. The cost associated with this approach is increased complexity of key generation and management. That is, several decryption keys must be communicated to the

receiver via an auxiliary communication channel, and synchronization between the transferred keys and the appropriate segments of the content must be maintained or regenerated at the receiver.

Attack analysis on a system containing watermarks shares certain similarities with the crypto-system analysis described above. For example, if an attacker is able to recover the secret embedding or extraction parameters (i.e., the stego key) of the watermarking system, he/she may be able to insert new watermarks or remove or jam existing watermarks. There are notable differences between the stego key in the context of watermarking systems and the encryption key described above. A detailed description of the stego key will be provided in the subsequent sections but for now the stego key can be assumed to comprise all the information necessary to embed and/or to extract a watermark from a host content.

The concept of security in watermarking systems also takes on a different meaning than what is typically envisioned by encryption systems. The main purpose of encryption is to prevent deciphering of a message without the presence of an encryption key. The main design goal of a watermarking system, in the context of present invention, is to thwart unauthorized removal or jamming of the embedded watermarks. These types of attacks typically involve obtaining information about the watermarking algorithm and its parameters (i.e., determining the stego key) and subsequently removing or jamming the embedded watermarks without degrading the perceptual quality of the host content.

Stego Key – Watermark Embedding

The term steganography means “covered writing” and it is derived from Greek word “stego” that means roof or cover and “graphy” that means writing. In other words, steganography is a technique of hiding messages or signals within a host content. The hiding place is generally described through a “stego key” that is used both in embedding and in extracting procedures. Generally, it is assumed that the hiding algorithm is public, but the parameters used in a particular hiding process are secret, and comprise the stego key. This is also known as Kerckhoff’s principle, which states that the security of a cryptosystem shall not be based on keeping the algorithm secret but solely on keeping the key secret. This is based on the idea that eventually the algorithm will be discovered and peer review of the algorithm is only going to benefit the security, design and acceptance of the algorithm.

In the context of the present invention, the stego key comprises all relevant information that is used to embed and/or extract a particular watermark within multimedia content. According to this generalized definition, watermarking algorithms should also be considered part of the stego key. The Kerckhoff's principle is correct only if the technology selection is fixed and only one technology is implemented in extraction devices that are distributed to the public. According to the embodiments of the present invention, multiple distinct technologies are implemented in the extraction apparatus, and the selection among them in any particular communication attempt is kept secret, i.e. it is part of the stego key. For example, embedding and extraction devices may be equipped with at least one of spread spectrum, quantization index modulation, peak-amplitude modulation, echo hiding or a combination of these technologies.

However, it is not necessary for every embedder and/or extractor to contain all possible watermarking technologies. The objective is to keep results of an attack inconsistent, unpredictable, and confusing both to the attacker and to the user of pirated content. For example, it may be desirable to lead the attacker to believe that an attack is successful, and to distribute his results, and then to find out that this is not the case, resulting in the attacker's loss of credibility. Similarly, it may be desirable to make the pirates invest time and effort to find a procedure, or a 'black box' that enables piracy only to discover that it works only for special cases.

In the case of embedders, it may be advantageous to reserve some of the embedding technologies for future use, as attacks on already deployed watermarks grow more sophisticated and more successful. Generally, the selection of watermarking technologies in the embedder must be as flexible and as dynamic as possible. This could mean that the selection of particular embedding technologies could be done in accordance with the degree of success or failure of piracy attacks on already released content.

Similarly, not all watermarking technologies may be implemented in each extraction device. This would simplify implementation of extractors in consumer devices and reduce silicon area and processing load. It will also make extractor performance unpredictable, as extraction results would differ from one device to another, making attacks inconsistent and frustrating to casual pirates.

It is important to note that different watermarking techniques may have different robustness profiles, e.g. one technology may produce better immunity to lossy compression and corruption by noise, while another may exhibit superior immunity to synchronization attacks. Therefore, by embedding and extracting watermarks using multiple algorithms and technologies, one can increase the overall range of distortions and attacks that can be tolerated (i.e. it may require multiple simultaneous distortions in order to disable watermark detection). Multiple technologies may require a multidimensional search for a weakness, while a single technology may require a one-dimensional search. Exact selection of the technologies that should be bundled together depends on the individual features of each technology. Generally, it is desirable to combine together disparate technologies with non-overlapping security weaknesses. For example, watermarks embedded using time-domain, spread-spectrum technology, which may be prone to synchronization attacks, may coexist with replica modulation watermarks to alleviate this weakness. It should be also noted that simultaneous embedding of watermarks with multiple technologies complicates differential analysis and may mislead and confuse the attackers. For this purpose, it is desirable that watermarks generated by different algorithms overlap in time, space, and frequency domains.

In many watermarking applications of the prior art, multiple watermarks are embedded using the same stego key but at different time intervals. This redundant embedding of watermarks is typically done to improve the chances of correct detection of watermarks. For example, the stego key for a spread spectrum watermark may comprise the seed for a random number generator. In such systems, the same seed may be used to repeatedly embed the same watermark throughout the content. In the system of present invention, different watermarking technologies are employed during different time intervals so that the time interval for a particular embedding is also considered to be part of a stego key.

In addition, in typical watermarking applications of the prior art, watermark strength is decided solely based on a robustness vs. transparency tradeoff. In the system of the present invention, watermark strength may be incorporated as a security feature of the system since the success of many blind attacks, as well as jamming and overwriting attacks, strongly depends on watermark strength. Therefore, in the context of the present invention, the watermark strength is also part of embedder stego key. Note that for many watermarking technologies, such as spread spectrum or replica modulation, the watermark strength does not affect the

extractor operation (other than changing the robustness of detection in the presence of channel impairments), while for others, such as peak amplitude modulation or distributed feature quantization, extraction parameters do change in accordance with the strength of embedded watermarks. In the latter cases, changing the embedding strength must be conveyed to the extraction device as part of the extractor key set.

Selection of watermark strength certainly affects the transparency requirement, i.e. the stronger the watermark, the more perceptible is its presence. However, the perceptibility of watermarks also depends strongly on the masking capability of the content itself. For example, content that is more noise like, such as recording of a live rock concert, may be more tolerant to increased watermark strength than a more structured content, such as string quartet music. Similarly, for still and moving images, it may be preferable to embed watermarks in regions within a certain range of luminance/chrominance values or spatial frequencies. Therefore, variations of watermark strength should be made in tune with content properties, either by some automatic method, or by human selection. In any case, the algorithm for varying watermark strength may be kept secret so that an attacker cannot determine these variations by analyzing the content.

The strength of embedding can also be modified in accordance with the value of the content that is being protected and the application of the content. In some cases, the additional distortion introduced in the content due to stronger watermarks may be negligible compared to the distortions introduced by the transmission channel (e.g., FM or AM channel broadcasts). In such cases, increasing the strength of embedding may be acceptable. In other cases, such as theatrical or DVD releases of motion pictures, higher fidelity requirements may not allow such overall increases in watermark strength. Alternatively, the strength of watermark may be locally adjusted for only portions of the multimedia signal of special interest. These portions may comprise climactic scenes or key audio portions in a movie or may be randomly distributed throughout the multimedia content.

In the presence of watermark strength variations, the attacker has two options. The first option is to apply enough distortion and/or noise throughout the content, which would guarantee that even the strongest watermark is removed. For a properly designed watermarking system, this action is expected to degrade the quality of the content beyond acceptable levels. It should be noted that the amount of distortion and/or noise required to remove the strongest

watermarks may vary from content to content. The strength of embedding can also be adjusted dynamically in accordance with the degree of success or failure of piracy attacks on previously released material. For example, future releases of a multimedia content may be embedded with stronger watermarks to make any previously developed attacks ineffective.

Alternatively, an attacker may try to identify the watermark strength in each particular content, and apply just enough distortion to each of them. This is a very difficult task in the presence of multiple technologies and other security features disclosed herein. Many of the security features disclosed below are designed to prevent this type of analysis.

The watermarking stego key of the present system is comprised of many components where each component may be considered one dimension of an n -dimensional stego key space. The constituents of the n -dimensional space may vary depending on the nature and type of the host content or the specific application of watermarking system. For example, the number and type of stego key components for still image watermarking may differ from those corresponding to video or audio watermarking. In addition, the stego keys for embedding and extraction are generally of different lengths and may span different dimensions in the stego key space. In general, an embedding stego key of length L_e may be represented as:

$$L_e = \sum L_e^i = L_e^1 + L_e^2 + L_e^3 + \dots + L_e^n \quad \text{Equation (5)}$$

Where n is the dimension of stego key space and L_e^i represents the length of each component, i , for $1 \leq i \leq n$, as shown in Fig. 2. In accordance with an example embodiment of the present invention, for each embedding session, a stego key is selected and used for embedding the content. The selection of the embedding stego key can be a random selection process or may be done in accordance with a pre-determined selection scheme.

The stego key components for an audio watermarking system, for example, may be constructed using a 6-dimensional key space. Fig. 3 provides an example of an embedding stego key for an audio watermarking system. The first component of the stego key (Dimension 1) may represent the particular time slot within the multimedia content that is targeted for embedding. In this example, using 16 bits allows selection of 65536 different time slots. The second component (Dimension 2) allows the selection of one or more embedding algorithms. Components 3 and 4 (Dimensions 3 and 4) respectively represent frequency shift and PN

sequence lengths that are associated with each embedding algorithm. Component 5 (Dimension 5) corresponds to the number of distinct frequency segments in the content that is targeted for embedding. Note that the granularity of time and frequency components may vary according to the desired level of watermark capacity, security, transparency and robustness in accordance with system performance requirements and particular embedding algorithm. In addition, the duration of time slots and/or the spectral width of frequency bands may not be held constant throughout the content. These variations may be based on a pre-calculated variation pattern, a random variation pattern or may be in accordance with the characteristics of the host content. It is also entirely possible to have different time/frequency granularities for different embedding algorithms. Incorporation of such additional features would result in an even longer stego key.

Note that the stego key illustrated in Fig. 3 only provides an example of a 5-dimensional stego key space. In practice, some dimensions may be added (or removed) to achieve the desired system performance. The illustrated configuration allows embedding of the same time/frequency/spatial segment of the content with one or more watermarking algorithms. Of course, the number and strength of overlapping watermark layers/bands must not produce perceptible artifacts as mandated by the transparency requirements of the system. System performance requirements greatly vary from one application to another. Similar tables may also be constructed for other forms of multimedia content such as texts, still images or video streams. For example, a watermarking stego key for a feature film may comprise additional spatial and temporal frequency dimensions corresponding to video portions of the motion picture, as well as spatial extent of watermarks corresponding to individual frames of the motion picture.

Computational efficiency may also play a role in selecting the extent of embedding stego key, especially in real-time applications such as embedding of the content right before broadcast, embedding during tape-to-tape transfers, data-to-film transfers or rip-embed-transmit procedures. In such cases, it may be required to select the extent of stego key in accordance with cost/time limitations of the embedding application. For example, only a limited set of embedding opportunities may be utilized or a limited number of embedding algorithms may be selected. The limitations due to current technical constraints, however, are expected to be less prohibitive in the future as improvements in speed/cost of computational resources take place.

Although multimedia content management systems based on watermarks are not generally anticipated to produce cryptographically secure systems, it is instructive to compare the stego key analysis of the present watermarking system with the encryption key analysis described above. Most encryption key space attacks involve trying different permutations of a binary sequence until the encryption key value is revealed. Similar brute force attacks on the stego key space of the present watermarking system involves more than manipulation of a binary sequence. In fact, attacking the watermarking stego key involves attacking multiple binary sequences in n different dimensions where each attack may involve a different set of computationally expensive operations, such as filtering, convolution, correlation, etc. The complexity of attacks is believed to be sufficient to thwart and discourage unauthorized usage of the watermarked content.

Fig. 4 shows a block diagram of an Embedding Apparatus 500 in accordance with an exemplary embodiment of the present invention. The incoming host signal 501 containing the digital host content is received by a receiver or other device incorporating a receiver (e.g., Embedder Reception Device 510 of the Embedding Apparatus 500). As the input host content signal 501 may be in a variety of formats and may comprise several audio, video, multimedia, or data signals, it is necessary for the Embedder Reception Device 510 to appropriately condition the incoming host signal 501 into the proper form that is recognizable by other components of the embedding apparatus 500. This conditioning may comprise signal processing steps, such as, for example, demodulation, decompression, de-interleaving, decryption, descrambling, resampling, A/D conversion, re-formatting, filtering, or the like. It is also understood that some of the required signal conditioning steps may be carried out in other sections of the embedding apparatus such as the Watermark Embedding Device 550. The conditioned (or partially conditioned) signal is then processed by the Identification Device 520 in order to identify multiple embedding opportunities or locations within the host signal. All possible embedding opportunities may be identified. Alternatively, the identification of the embedding opportunities may be performed in accordance with all or some of the embedding technologies that may be used for embedding watermarks. A Selection Device 530 then selects a subset of the identified embedding opportunities.

An optional Embedding Technology Storage Device 540 may be provided in order to store available embedding technologies. The Storage Device 540 may be regularly upgraded to

contain up-to-date versions of the embedding technology parameters, algorithms or settings. It should be understood that the presence of a separate storage device may not be necessary, as other components of the embedding apparatus such as the Selection Device 540 or the Watermark Embedding Device 550 may contain the appropriate information related to the available embedding technologies and/or contain upgradeable memory modules that can be utilized for this purpose. The Selection Device 540 may also select one or more watermark embedding technologies from the Storage Device 530 (or other storage location). Once the appropriate embedding opportunities and the one or more watermark embedding technologies have been selected, the Watermark Embedding Device 550 embeds the watermarks in accordance with the selected watermark embedding technologies at the locations corresponding to the selected subset of embedding opportunities in the host content to produce an embedded host signal 560. The embedded host signal 560 may then be further processed, stored or transmitted.

The digital host content contained in the incoming host signal 501 may comprise one of multimedia content, audio content, video content, audiovisual content, image content, or the like.

The Selection Device 540 may select the subset of embedding opportunities to provide at least one of optimum robustness, optimum security and optimum transparency of the watermark. Further, the Selection Device 540 may select the subset of embedding opportunities to provide a desired tradeoff between levels of robustness, security, and transparency of the watermark.

A plurality of watermarking embedding technologies may be selected from the Storage Device 530 by the Selection Device 540 and used by the Watermark Embedding Device 550. For example, at least two different watermarking embedding technologies may be selected and used by Watermark Embedding Device 550.

The embedding opportunities may be identified by the Identification Device 520 in accordance with characteristics of the host content. These characteristics may comprise at least one of temporal frequency, spatial frequency, duration, peak amplitude, luminance, chrominance, masking capability of the content, or the like. The embedding opportunities may also be identified in accordance with parameters of the watermark embedding technologies. These parameters may comprise at least one of autocorrelation delay, frequency shift, PN

sequence, quantization index, strength of embedding, chip rate, embedding bit rate, or the like. The embedding opportunities may be identified in accordance with both characteristics of the host content and parameters of the watermark embedding technologies.

The embedding opportunities may comprise a multi-dimensional space. The dimensions of the space may be comprised of at least two of: temporal frequency band, spatial frequency band, time segment, spatial extent, time delay, frequency shift, PN sequence, or embedding algorithm type.

The multiplicity of watermarks may be embedded by the Watermark Embedding Device 550 using the same selected watermark embedding technology but with different embedding parameters. The extent of the selected embedding opportunities may be determined in accordance with at least one of desired transparency, robustness, or security of the system. A different subset of the identified embedding opportunities may be selected by the Selection Device 540 for embedding each host content. The different subsets may have no common elements. Alternatively, the different subsets may have at least one common element.

Further, different subsets of embedding opportunities may be selected by the Selection Device 540 for embedding different copies of the same host content (as discussed in detail below in connection with Fig. 10). The different subsets used for the different copies of the host content may have no common elements. Alternatively, the different subsets may have at least one common element.

The subset of embedding opportunities may be selected by the Selection Device 540 in a random or pseudo-random fashion.

A different number of embedding opportunities may be produced for different implementations of an embedding device 500. A pattern of selected embedding opportunities may uniquely identify each embedding device. Further, a pattern of selected embedding opportunities may uniquely identify each embedding.

The subset of the identified embedding opportunities may be adaptable in response to observed or anticipated attacks. For example, the selecting of the subset of the identified embedding opportunities may comprise selecting (e.g., by the Selection Device 540 or other component of the Embedding Apparatus 500) at least one type of attack available against the selected watermark embedding technology, and producing a first set of embedding

opportunities that may be adapted to be transformed into a second set of embedding opportunities in the presence of the attack.

It should be understood that the Embedding Apparatus 500, as shown in Fig. 4, may comprise a variety of digital, analog, optical or acoustical components. For example, the Embedding Apparatus may be implemented using a digital signal processing (DSP) unit, FPGA and ASIC devices, or may be implemented in a computer or hand-held device. It should also be understood that while the Embedding Apparatus 500 of Fig. 4 may be implemented as a single embedding unit, it is also possible to break-up its constituent components to form a distributed embedding device. For example, it is entirely possible to place the Watermark Embedding Device 550 at one physical location while the remainder of the embedding apparatus is placed at another physical location or multiple physical locations. The distribution of the embedding components may be done in accordance with the computational requirements of each component and the availability of computational resources at each location. The various components of such distributed apparatus may be interconnected using a variety of connectivity means, such as, for example, the Internet, dedicated phone lines, various wired or wireless computer networks, or even physical media such as portable storage devices.

Stego Key – Watermark Extraction

In order to successfully extract the embedded watermarks, an extraction stego key set must be present at the extraction device. The communication of the extraction key set is usually accomplished by permanently storing the extraction key set in the receiver apparatus, or communicating the extraction stego key set via an additional communication channel (e.g., smart card, internet connection, phone line, etc.), or carrying the extraction stego key set via the same communication channel (e.g., as part of the file header, on non-data carrying portions of CD's and DVD's, as part of the embedded watermarks, etc.). The disadvantage of the systems where the key set is permanently stored at the receiver is in their vulnerability to key space attacks. That is, if the extraction key set is stolen or revealed, the entire system may be compromised. This is especially true if a large number of extraction devices with the same extraction key set have been distributed and security updates are not feasible (e.g., in consumer

electronic devices such as CD and DVD players). Key delivery using an auxiliary channel, although proven effective in certain cases such as set top box applications, have the disadvantage of relying on an additional communication channel. Such auxiliary channels may not be cost effective, may be susceptible to interception and jamming and may not be available in certain applications (e.g., analog recording devices).

Before describing the details of key exchange and key management of the present watermarking system, some characteristics of the extraction technique must be described. The present system, in most general cases, uses an asymmetric stego key approach for extracting the watermarks. In the absence of channel distortions, where content is delivered in pristine condition to its destination, it may be sufficient to know the exact embedding algorithms and the specific embedding parameters in order to extract the watermarks from the received content (i.e., use a symmetric stego key approach). However, in the presence of channel distortions, which may be the result of intentional or unintentional processing or attacks, the exact version of the embedding stego key may not be successful in extracting the distorted watermarks. This is due to the fact that the embedded watermarks may appear to have been embedded with a different set of parameters. For example, for a watermark that is hidden in an audio signal in the frequency band 4,000 Hz to 4,200 Hz, a pitch-shift processing that increases pitch by 5%, would move the embedded region to the band 4,200 Hz to 4,410 Hz. A properly designed extractor must search for the presence of watermarks not only in the original 4,000-4,200 Hz band but also in 4,200-4,410 Hz band (or, approximately, in 4,200-4,400 Hz band) in anticipation of probable channel impairments. Similarly, for an image with an 8x8 watermarked pixel area, if the screen size is transformed from 16:9 into 4:3 aspect ratio, the 8x8 watermarked pixel area would become an 8x6 pixel area. An extraction stego key set that is configured for both the 8x8 and 8x6 pixel areas would successfully extract watermarks in the presence (or absence) of this type of distortion.

In the system of present invention, the embedder key space may include as many keys as possible to account for all possible transformations. However, it is not necessary to incorporate all possible transformations of the stego key into the embedder key space. Furthermore, the extractor key space is not necessarily limited to embedder key space. It may not only include the entire embedder key space, so that each embedded watermark has a

chance of being detected in clear channel, but it may also include a substantial number of stego keys that are generated by transformations outside embedder key space.

This concept of asymmetric embedding and extraction key sets is shown pictorially in Figures 5a and 5b. In Fig. 5a, the small circle represents an embedding stego key set and the large oval represents a set of extraction key values. The fact that the embedding key set completely lies within the larger extraction key set indicates that all embedded watermarks (in the absence of channel distortions) may be extracted from the content. This also indicates that some extraction attempts (the ones that look for embedded watermarks outside the embedded regions) are necessarily futile but this additional processing is a small price to pay for achieving enhanced security and improved robustness performance. In Fig. 5b, the same set of extraction keys are used for extraction in the presence of channel distortions that have altered the appearance of the embedding key set. As illustrated in Fig. 5b, the distorted embedder key set may fall partly outside the extractor key set that is utilized in a particular extractor, indicating that it is not necessary to capture all watermarks. In general, in the presence of channel distortions some portions of the embedder key set may remain unchanged or stay within the original embedding key space (i.e., overlap area between the original and distorted key sets in Fig. 5b) while other portions of the embedder key set may move to the areas outside the original embedding key space but within the extractor key space. Yet, other portions of the embedder key set may completely move outside the extractor key space. This illustrates the fact that it is possible and advantageous to select the embedder key set in such a way that it includes many 'transformed' stego keys (i.e., the stego keys that can be converted from one to the other through regular signal processing or through an attack). In addition, this figure illustrates that having a larger extractor key space produces an increased chance of successful watermark detection in the presence of distortions. Finally, this figure illustrates the fact that it is not necessary to make the extractor key space large enough to cover all possible transformed embedder keys. It is quite acceptable to have some watermarks lost due to signal processing and/or attacks. The selection of the extractor key space is governed by the tradeoff between probability of stego key being transformed into a certain region, and the burden on extractors (in terms of processing power and probability of false detections) incurred for searches in those regions in accordance with the security requirements of the system.

Stego Key Design

One of the goals of the present watermarking system is to embed each multimedia content with a different stego key set, not to communicate the specific embedding key set to the extraction device, and yet be able to extract the embedded watermarks in the presence of potential channel impairments with acceptable reliability. Two factors facilitate the design of such system. First, there are typically many opportunities for insertion of watermarks in most multimedia content (e.g., a typical movie is 2 hours long whereas a typical audio watermark is only a few seconds in duration). Second, in many applications of watermarking systems it is not necessary to extract all embedded watermarks. In many cases, such as content management applications, the extraction of only a few embedded watermarks produces the desired outcome. On the other hand, in order to for an attacker to defeat the system, all or most embedded watermarks need to be removed from the content. Furthermore, in order to develop a successful attack that can be distributed to unsophisticated users, the attack must be successful for substantially all types of content that utilize the watermarking system, including the presently available content and future releases. The combination of these factors are advantageously used in the present system to eliminate the need for exchange of watermarking extraction keys and to achieve a security level that approaches cryptographic systems.

One way of achieving this goal is to embed a content with at least one embedding key set, selected at random or based on a pre-determined algorithm, from the set of all possible embedding keys and at the receiver device, perform an exhaustive search of the entire extractor key space. With the proper design of extractor key space, the probability of success can be made arbitrarily high. This task may be feasible if the length of the extractor stego key is small or only a small subset of all possible stego keys are utilized in the extractor. For longer stego keys however, such as the exemplary stego key illustrated in Fig. 3, this task is too computationally expensive for practical implementations. In addition, an exhaustive search may increase the rate of false watermark detections beyond an acceptable limit.

An alternative way of achieving this goal is to use every possible embedding stego key set for embedding the content and, on the extractor side, perform at least one extraction attempt, randomly selected from the embedder key space. This approach would indeed simplify the task of watermark extractors, but at the expense of compromising content quality and embedding complexity. Therefore, it is apparent that the optimum design may be

somewhere in between - with multiple watermarks being embedded and multiple searches being executed at the extractor, but neither all embedding opportunities nor all extraction opportunities are exhausted. From the security point of view it is desirable to have as many embedded watermarks as possible and execute as many extraction attempts as possible. However, the number of embeddings and extractions attempts are limited by transparency and processing requirements of the system. Therefore, it is desirable to know the relationship between a selected number of embedding and extracting key sets, and the probability of successful detection, which will be the subject of analysis below. Further, from security point of view it is very important *how* those keys are selected, not only *how many*.

The concept of stego key selection is presented pictorially in Fig. 6. In Fig. 6a, four different embedding stego key sets belonging to four different embedding devices are represented as sets **A**, **B**, **C** and **D**. Fig. 6a indicates a non-zero intersecting area between all four keys. This is not a requirement of the system and is only presented in Fig. 6 to facilitate the understanding of the disclosed concepts; it is entirely possible to have embedding key sets that are non-overlapping and/or with no common intersection area. The presence of non-overlapping or part-overlapping embedding key sets indeed improves the security of the systems since if one key set is compromised other key sets may still be used.

Fig. 6a can also be used to illustrate key sets used by the same embedding device but for four different contents. The particular stego key set(s) assigned to each embedder, or the particular subset of such keys selected for each embedding session, may be done randomly or selected in a non-random fashion. For example, the assignment/selection of embedding stego keys may be done in different stages to maximize the lifespan of the deployed system; early embedders and embedder runs may use a limited set of watermarking technologies, and limited watermarking strengths, while later versions may deploy more sophisticated technologies and/or stronger watermarks that are tailored to thwart circumvention attempts which may have already been deployed by attackers. Similarly, one set of stego keys may be used for embedding relatively low-value content (e.g., a TV commercial) while another set may be used to embed a higher value content (e.g., a feature film).

In yet another variation of the above technique, the entire embedding stego key space may be partitioned into different sections and each section may be assigned to a different geographical region, to a different group of customers, or a particular type of multimedia. This

type of partitioning of the embedding key space would allow forensic tracking of the embedded content. For example, once a suspect content is received and examined for the presence of watermarks, the origins of the content may be traced back to the stego key (or the partition) that allowed successful extraction of the embedded watermarks. It should be noted that the above discussion does not preclude the existence of overlapping partitions (or sections) of the embedding stego key space. In such cases, the origins of an embedded content may be traced back by determining the particular stego key that produces the best detection results (e.g., produces the largest number of detections or the most reliable detections). This result would indicate the origins of the embedded content with maximum likelihood.

Fig. 6b represents the extraction key sets for three different extractor devices, **X**, **Y** and **Z**. These keys can be assigned at random to the corresponding extraction devices, but also can be assigned in view of extraction device properties. For example, if the extractor resides in a camcorder that may be used for theater piracy, the extractor key set doesn't need to include transform keys obtained through speed up or slow down of the content. Similarly, if the extractor resides in a software module that has an expiration date, upon which new software must be downloaded, then it would be advantageous to make phased distribution of extractor keys similar to that proposed for embedders.

The most important feature illustrated in Fig. 6b is that different extractors have distinct sets of extractor keys, so that if a single (or a few) of the key sets are compromised, other key sets may still be used. In addition, Fig. 6b can be used to illustrate key sets used by a single extractor in different runs. An important property in this case is that the behavior of the extractor is not deterministic, i.e. the outcome of different runs may not be the same. It should also be noted that in Fig. 6, all embedding stego key sets (as well as all extraction key sets) appear to be the same size. This is not necessarily true and is only done to facilitate the understanding of the concepts. In fact, in accordance with the present invention, it may be preferable for different embedding and extraction key sets to be of different sizes. Similar to the embedding stego keys, there is not a strict requirement for all extraction key sets to have an overlapping area but such overlapping areas may be necessary to produce the desired performance reliability. In addition, it may be desirable for each extractor to contain several sets of non-overlapping (and perhaps unused) extraction key sets to make it possible to

periodically switch to a new set of extraction keys or to permanently retire a particular set of extraction keys.

Fig. 6c shows the extraction key set, **Z**, that is capable of extracting watermarks that are embedded using any one of the embedding key sets **A** through **D**. Note that some portions of sets **A** and **B** reside outside the detection region of extractor **Z** and thus may not be able to extract all embedded watermarks in contents **A** or **B**. On the other hand, since watermarks are embedded redundantly throughout the content, such extraction failures may not be of any consequence. The stego key system architecture shown in Fig. 6 provides for each embedding to occur with a different stego key and each extraction device to contain a different set of extraction keys. As long as there is some overlap between an embedding key set and an extraction key set, watermark recovery is possible. Furthermore, a successful attack on one embedded content or one particular extractor does not result in successful removal of watermarks from other embedded contents using other extraction devices. Different embedding stego key sizes may be used to produce variable levels of transparency, security and robustness. For example, a content that is embedded redundantly with three different embedding algorithms may produce better security but lower transparency compared to a content that is embedded using one embedding algorithm. Similarly, an extractor set with a smaller key set may provide reduced robustness performance but better computational efficiency. The framework of the present architecture provides for a flexible system design structure where security, robustness, transparency and computational cost tradeoffs can take place. An example of how stego key length may affect different requirements is shown in the table of Fig. 7. This table is presented here to illustrate the complex inter-relationships between system requirements and system parameters. For example, according to the table of Fig. 7, assuming a fixed payload capacity, increased length of embedding stego key may produce a more secure system at the expense of reduced transparency.

The length of embedding and extraction stego keys, the number of different extraction key sets as well as the degree of overlap between the different extraction key sets can be used to make the present watermarking system optimally secure, transparent, reliable and computationally inexpensive. These parameters can be adjusted according to the value and type of the content that is being protected and the type of extraction device that is being used.

For example, a more secure implementation of extractors may be selected for hand-held camcorders (to guard against movie piracy in theatres) than for portable MP3 players.

Fig. 8 shows a block diagram of an Extractor Apparatus 600 in accordance with an exemplary embodiment of the present invention. The incoming embedded host signal 560 (e.g., produced by the Embedding Apparatus 500 of FIG. 4) is received at a receiver or other device incorporating a receiver (e.g., Extractor Reception Device 610 in the Extractor Apparatus 600). Similar to the conditioning operations discussed in relation to the Embedder Reception Device 510 of Fig. 4, the Extractor Reception Device 610 may appropriately condition the incoming embedded host signal 560. A Stego Key Selection Device 620 then selects at least one stego key from a collection of stego keys that are stored in Stego Key Storage Device 630. The selected stego keys are subsequently used by the Watermark Extraction Device 640 to recover the embedded watermarks from the embedded host signal 560 to provide the recovered watermarks 650.

The Stego Key Selection Device 620 may select the at least one stego key to produce at least one of optimum robustness, security, and computational efficiency for the extraction of watermarks embedded in the host content. Further, the Stego Key Selection Device 620 may select the at least one stego key to produce a desired tradeoff between levels of robustness, security, and computational efficiency for the extraction of watermarks embedded in the host content.

The Stego Key Selection Device 620 may select the at least one stego key in a random or pseudo-random fashion.

The collection of stego keys stored in the Stego Key Storage Device 630 may comprise one stego key, at least two stego keys, or pairs of stego keys. The collection of stego keys may further comprise at least a pair of stego keys with no common elements, or at least a pair of stego keys with at least one common element.

The presence and value of extracted watermarks 650 may be reported (e.g., to a particular user or device) at pre-determined time intervals.

The selecting of the one or more stego keys by the Selection Device 620 may be adapted in accordance with a desired false positive detection rate. The selecting of the one or more stego keys may be adapted to produce a desired probability of successful extractions. Further, the selecting of the one or more stego keys may be adapted to produce a desired

computational complexity for the extraction of the watermarks. Additionally, the selecting of the one or more stego keys may be adapted to anticipate transformations of the host content. Such transformations of the host content may modify watermark characteristics of the embedded watermarks. For example, the transformations may alter the appearance of at least one watermark that is embedded with a first embedding stego key such that the at least one embedded watermark appears to have been embedded with a second embedding stego key.

The size of a selected stego key may vary from one host content to another. A size of the collection of stego keys may vary in accordance with a type or value of the host content. Further, the size of a selected stego key may vary in accordance with a type or value of the host content.

The collection of stego keys in the Storage Device 630 may comprise a subset of all possible extraction stego keys. The collection of stego keys may be adapted in accordance with an observed or anticipated attack. Further, the collection of stego keys may be adapted in accordance with an expiration date.

Extraction of a subset of all originally embedded watermarks may be enabled by the Watermark Extraction Device 640 in accordance with the at least one stego key. Alternatively, extraction of all possibly embedded watermarks may be enabled in accordance with the at least one stego key.

The Extractor Apparatus 600 may be implemented using the same or similar technology as the Embedding Apparatus 500 discussed above. Further, like the Embedding Apparatus 500, the Extractor Apparatus 600 may be implemented as either a single unit or as a distributed device consisting of several discrete components at the same or different physical locations.

It should also be appreciated that the Embedding Apparatus 500 described in connection with Fig. 4 may be used in connection with the Extractor Apparatus 600 described in connection with Fig. 8 to form a system for embedding and extracting digital watermarks.

Presence of Additional Watermark Layers

The disclosed embedding and extraction concepts are equally applicable to many watermarking applications that require additional watermark layers. The presence of these additional layers may be used, for example, to extend the payload capacity of the existing

watermarks, provide tiered security (and access) architecture or provide forensic or transactional capabilities. For example, a multimedia content may contain both a copy control watermark, with a 3-bit payload, and one or more forensic or transactional watermarks with 50-bit payloads. The payload of the second set of watermarks may be used to carry identification, ownership and/or origination information, including, but not limited to, the time and date of origination, purchase or distribution of the content, the name or credentials of the recipients (e.g., movie theatre, retail store, individual, etc.) or the time of date of presentation of the content. Such information may be later used to unravel the origins and distribution paths of the multimedia content.

Furthermore, the second set of watermarks may be used to convey information regarding the identity and authenticity of the transmission channel of the host signal. The transmission channel, in general terms, may comprise any origination, storage, or transmission device, channel, or medium, such as physical storage media, devices that are used to transmit, broadcast or relay the multimedia content, and the like. Examples of the identity information of the transmission channel may include ID or serial numbers that reside on today's most optical and magnetic memory devices, IP addresses, URL's, email addresses, or any other information that uniquely identifies all or part of the transmission channel components. In order to prevent fraudulent creation of such identification information, cryptographic or non-cryptographic techniques may be used to ensure authenticity of the embedded information. Examples of these techniques include, but are not limited to, the use of digital signatures, hash functions, digital certificates, encryption algorithms, and the like, which may or may not include the participation of a trusted third party. These and other authentication techniques are well known to those skilled in the art. The identification information carried within a second set of watermarks may be extracted and authenticated at a reception device; if authenticity is successfully verified, access/copy control information that resides within the first set of watermarks may be extracted and acted upon. This could include allowing the user to playback, record or transmit the content. Using this technique, any content lacking proper credentials may not be accessible or may be subject to a different set of usage rules.

Design Guidelines and Examples

Example 1:

Some of the concepts of the present invention may be illustrated using the following example. This example provides embedding of a monophonic sound track of a 2-hour motion picture. One simple method of embedding would be to first divide the sound track into a finite number of time-frequency segments and to then embed watermark bits into randomly selected time-frequency segments (the details of how the actual watermark bits are embedded are not very important for this example, as any one of several algorithms described in the systems of prior art may be utilized). This technique, which resembles a Frequency Hopped Spread Spectrum technique, is shown in Fig. 9. In this figure, the specific time-frequency bins containing watermark bits are highlighted.

In order to further illustrate some of the features of present invention using this example, the following parameters may be assumed:

- Available bandwidth for embedding = 16 KHz
- Frequency bin size = 200 Hz
- Number of distinct frequency bins (hops) = $16,000/200 = 80$
- Total length of audio track = 120 minutes
- Time slot size (i.e., one bit duration) = 20 ms
- Number of distinct time slots = $120 \text{ (min)} \times 60 \text{ (sec/min)} / 20 \text{ (ms)} = 360,000$
- Number of distinct time-frequency segments = $360,000 \times 80 = 28,800,000$
- Number of bits in each Watermark packet = 100 bits
- Number of watermark packets that can be accommodated = 288,000

Example 1: Embedding

Watermark packets may be embedded into one or more of the possible 288,000 available locations. In theory, it suffices to embed in only one location and subsequently detect the embedded watermark with an extractor device. But in practical systems, usually a large number of watermark packets are embedded into the content. This is done to improve the reliability of detections in the presence of, for example, channel impairments, intentional attacks or standard multimedia processing steps which may interfere with the detection of embedded watermarks. In addition, not all potential embedding locations may be suitable for

the placement of watermarks. For instance, due to the nature of the multimedia content and the particular embedding algorithm, certain locations within the content may not be able to meet the transparency/robustness/security requirements of the system. In such cases, the embedding of watermarks at certain locations may have to be skipped altogether or be done at reduced strength levels.

According to the present invention, each copy of a particular multimedia content may contain a different number of watermark packets that are embedded in different locations. In the context of present example, there are 288,000 possible locations within the audio track that can accommodate the embedding of watermarks. Fig. 10 provides an example of embedded watermark locations for 6 audio tracks. Specifically, content #1 contains 44 watermarks that are embedded in locations 11271, 13809, ..., 268986, content #2 contains 45 watermarks embedded in locations 11271, 14063, ..., 278485, etc. Examination of Fig. 10 reveals that only 159 unique watermark locations have been used to embed contents 1 through 6, with exactly 20 common locations between all embeddings. The underlined numbers in Fig. 10 correspond to watermark locations that are common between all 6 audio tracks. As stated above, there is no requirement for having common embedding locations between all embedded content; their presence in the present example merely facilitates the understanding of the underlying concepts.

The different contents listed in Fig. 10 may represent 6 different embeddings of the same audio content (e.g., prepared for distribution of a movie to customers) or 6 distinct audio tracks (e.g., 6 different movies). The locations of embedded watermarks shown in Fig. 10 were selected pseudo-randomly using a random number generator, although it may be desirable to select some or all embedding locations deterministically to be able to meet robustness/security/transparency requirements of watermarking system for each particular content.

Example 1: Detection

One detection technique may be to examine all 159 locations (or even all 288,000 possible locations) in search of the embedded watermarks. Another detection technique may be to examine only the 20 common locations in search of the embedded watermarks. A preferred technique, however, is to search an arbitrary number of watermark locations which includes all

or some of the “common” locations. This approach is illustrated in Fig. 11. The underlined search locations in Fig. 11 comprise a subset of “common” embedding locations previously discussed (recall that 20 common watermark locations existed between the 6 embedded content; the extractors of Fig. 11 contain at most 9 of these locations). Thus, in the absence of major channel impairments that interfere with the detection of watermarks, the extractor devices of Fig. 11 are guaranteed to detect embedded watermarks from all 6 content.

In addition to the common detection locations, there are several other locations within each extractor set, shown with ***bold-italic*** font in Fig. 11, that produce detections from at least one embedded content. There are also several other locations within each extractor (excluding extractor #4) that produce no detections from any of the embedded content. These locations, however, may produce detections from one or more of the embedded content in the presence of channel distortions or intentional processing. It has to be noted that these particular extractors only provide an example embodiment of the present invention and other variations are certainly possible. In particular, all 5 extractors of Fig. 11 are shown to have a number of common elements (i.e., the underlined numbers). This, in general, is not a requirement of the present invention as it suffices for each detector to have at least one location in common with each embedded content.

The extractors, numbered 1 through 5, may represent five different detection devices or five different sets of detection stego keys contained within a single extractor device. In the latter case, the extractor device may be configured to do one or more of the following:

1. Always search the same detection locations (e.g., the locations labeled “extractor 1”). This technique simplifies the design of extractors but it may not provide adequate security since breaking of a single stego key would render that extractor useless.
2. Search one set of detection locations (e.g., the locations labeled “extractor 1”) but switch to a different set of locations after a pre-defined period of time (e.g., 1 year later). The switching of stego keys may occur periodically, randomly and/or repeatedly among all stego key sets available to the extractor.
3. Search one set of detection locations (e.g., the locations labeled “extractor 1”) but switch to a different set upon system request (e.g., in case “extractor 1” locations are compromised and are no longer secure). Alternatively, such instructions to switch to a

new set of stego keys may be conveyed through the data carried within the embedded watermarks or a different layer of embedded watermarks.

4. Switch between different set of search locations (e.g., randomly select a set of locations at system startup or each time a new content is presented). Such switching may occur in accordance with a uniform or non-uniform probability distribution. The switching may further occur to produce detections in accordance with a new detection algorithm, a new frequency band, a new time slot, an new location with particular masking capabilities or a new location with particular value (e.g., climactic scenes, etc.).
5. Always search using more than one set of search locations.

The above list is not intended to be comprehensive. There are certainly other variations, within the scope of the present invention, that can be readily implemented by a person skilled in the art.

One of the advantages of using the embedding and detection techniques illustrated by this example is that if a single content is attacked and modified in a way that all embedded watermarks are obliterated, the same modifications are not likely to remove all embedded watermarks from a second content. Similarly, circumvention of a single extractor device is not likely to significantly affect the proper operation of all other extractors. This is further illustrated in Fig. 12 where it is assumed that all watermark locations contained within extractor #3 have been compromised (e.g., an attacker has been able to determine how to circumvent the detection of watermarks from all locations listed under extractor #3). The effect of this attack on the remaining extractors is shown in Fig. 12 by using grayed-out numbers to highlight the compromised locations. Particularly, it can be seen that only 4, 7, 5 and 3 locations in extractors 1, 2, 4, and 5, respectively, are compromised. These extractors are still perfectly capable of detecting embedded watermarks from the majority of their constituent search locations.

There are additional features of the present invention that can be illustrated using this example. Particularly, the extractors of Fig. 11 provide different levels of detectability for different embedded content. Specifically, careful examination of search locations in extractor #1 reveals that this device is configured to detect 21 embedded watermarks from content #1 but only 9 embedded watermarks from content #2. This approach produces different levels of detectability (i.e., robustness) for different content-extractor combinations and adds further

uncertainty to the exact configuration of embedding/extraction devices. The amount of detection variability can be randomly distributed among all content-extractor combinations or can be tailored to produce desired levels of robustness/security for a particular content-extractor combination. In general, by adjusting the relative number of embedded locations to the number of search locations, desired levels of system robustness, transparency and security may be achieved. Design guidelines for making such adjustments will be presented below.

Additional sets of search locations may also be incorporated in the extractors to account for future attacks on the system. The idea is that if, and when, a set of watermark extraction locations are compromised, those locations may be permanently retired and replaced by a new set of “alternative” embedding locations. The alternative locations may be completely different from the original locations or may contain a few retired locations to retain backward compatibility with the existing multimedia content. Extraction devices that are equipped with external communication capabilities may be easily updated with the new set of search locations. However, the majority of extractors are likely to be operating with no connectivity. Such devices may initially contain both the original and the alternative set of search locations and continually search for all embedded locations (even though no embeddings are yet present in the alternative locations). Another option would be for the extractor to automatically switch to the alternative set of locations after a pre-defined period of time (e.g., switch to a new set of locations every 6 months).

The above example only illustrates a portion of the capabilities and features of the present invention. For instance, in the above example only a single watermarking algorithm was considered. In accordance with the broader scope of the present invention, different embedding algorithms may be used to simultaneously embed different watermark packets into the host content. These packets may overlap with each other in time, frequency or space. In addition, while this example only considers a monophonic audio signal, the general embedding technique may be extended to include multi-channel audio signals, video signals and still images.

Design Guidelines:

The following parameters and conventions may be used for the development of generalized design concepts that follow. The assumption is made that the k^{th} embedder has inserted E_k watermarks in a particular content using an embedding key set. It is also assumed

that there are X_0 stego keys in the extractor key space, and the j^{th} extractor selects X_j keys among them randomly in a particular run, with uniform probability distribution. It is further assumed that when the embedder and extractor keys match, there is a probability, $0 \leq p \leq 1$, of a successful extraction. The probability of n successful extractions can be calculated according to Binomial distribution:

$$P(n) = \binom{X_j}{n} (pE_k / X_0)^n (1 - pE_k / X_0)^{X_j - n} \quad \text{Equation (6)}$$

Equation 6 governs the tradeoff between the number of embedded watermarks, E_k (which is related to the transparency of the watermarks), the number of extraction attempts, X_j (which is related to the processing load of the extraction), and the number of stego keys, X_0 (which is related to the security of the system and robustness of detection). For example, for a design objective that dictates a probability of no extractions equal to 10^{-6} , the necessary number of extraction attempts may be calculated, by setting $n = 0$ in Equation 6, as:

$$X_j = \frac{-6 \ln(10)}{\ln(1 - pE_k / X_0)} \approx \frac{13.8 X_0}{pE_k} \quad \text{Equation (7)}$$

It can also be readily shown that the expected number of extractions is:

$$E(n) = pX_j E_k / X_0 \quad \text{Equation (8)}$$

By substituting X_j obtained from Equation 7 into Equation 8, it can be seen that for the above example approximately 13.8 extractions per content are expected.

If it is further assumed that the attacker knows all possible stego keys (e.g. through a disgruntled employee disclosure), this would enable the attacker to design a 'black box' that would overwrite all watermarks with a garbled message. In this case, the attacker would insert $X_0/E_k = (pX_j/13.8)$ times more watermarks than the original embedding. In a properly designed watermarking system, this should produce unacceptable levels of perceptual artifacts. In order to illustrate this effect, Equations 6 through 8 may be applied to the example of a copy control system for movies based on audio watermarks. If a movie lasts 90 minutes, and the k^{th} embedder inserts, on average, one watermark per second of multimedia content, then $E_k = 5,400$. Based on the current assessment of extraction devices and audio watermarking technologies, it is anticipated that a 5-MIPS extractor can search about 20 watermarks per second, i.e. $X_j = (20 \times 90 \times 60) = 108,000$. In the presence of a sophisticated distortion attack, a probability of success of about $p = 0.25$ may be achieved. Using Equation 8, the calculated

value of the extractor key set size may be obtained as $X_0 = pX_jE_k/13.6 \approx 10^7$ stego keys. If an attacker were to overwrite all embedded watermarks, he/she would generate a content distortion equivalent to $X_0/E_k = 1,800$ times the distortion of a single embedder run. This is equivalent to having a jamming-to-signal ratio that is 32 dB worse than watermark-to-signal ratio, which is expected to produce unacceptable signal quality.

In many applications, such as copy management of multimedia content, the success criterion may be specified as extracting at least one watermark in the entirety of a multimedia content. In order to calculate the probability of this occurrence, it is easier to calculate the following: $P\{\text{success}\} = 1 - \text{Probability}\{\text{no successful extractions}\}$.

Increasing the probability of success (i.e., decreasing the probability of no extractions) results in only modest increase in the value of X_j . This relationship will be explored in detail below.

As a side note, assuming an independent watermark extraction process for now, it can be shown that the Binomial probability distribution can be approximated by a Poisson distribution if the following two conditions are satisfied:

- 1) $X_j \gg 1$ and 2) $(pE_k/X_0) \ll 1$ such that $E(n) = \mu = pX_jE_k/X_0$ is finite

In the case of the present watermarking system, both conditions are satisfied. Therefore the probability distribution of the extracted watermarks may be approximated by:

$$P(n) = (1/n!)(e^{-(pX_jE_k/X_0)})(pX_jE_k/X_0)^n \quad \text{Equation (9)}$$

Calculations involving the Poisson distribution are simpler than the ones for Binomial distribution. It can be shown that the mean and variance of Binomial distribution are equal to each other, and using the parameters described thus far, they are given by the expression (pX_jE_k/X_0) .

It is now instructive to look at some limits involving the parameters of the probability functions described above in order to devise further system design guidelines.

The probability p represents the likelihood of extracting a watermark independent of stego-key distribution. The value of this probability, among other factors, depends on the audio content, the strength of watermark embedding, the embedding technology and the amount of channel impairments. This probability should be close to 1 for an embedded content with no or little impairments. It is not expected to encounter a value much smaller than $p=0.1$ for an embedded audio content with reasonable quality.

The ratio $r_j = (X_j/X_0)$ is related to the security of the extractors. That is, a smaller r_j produces a more secure system. This ratio is always smaller than or equal to 1, as X_j is a subset of the larger set X_0 . However, it is desired for X_j to be as large as possible to ensure sufficient overlap with any embedder stego key space. As described above, the limiting factor for the size of X_j is processing load of the extractors; a value of $X_j \sim 10^5$ may be reasonable for a typical motion picture content and today's average processing capabilities. The value of X_0 must be large enough to provide sufficient security. For example, $X_0=10^8$ produces $r_j=10^{-3}$. This allows 1,000 distinct extractor key sets to be distributed. Note that even in the limiting case where $X_j \sim X_0$, the extraction system can be made arbitrarily secure for large values of X_j .

E_k represents the number of different watermarks embedded in the k^{th} content. E_k is a subset of E_0 , the total number of possible embedding key sets. In the limiting case, where $E_k \sim E_0$, each content is embedded with all (or close to all) possible watermark combinations, and thus regardless of the ratio r_j , a watermark should be extracted with high degree of likelihood. As described above, $X_0 \geq E_0$ in order to accommodate possible channel distortions and key set transformations.

Returning to the analysis of Equation 7, it may be assumed that an arbitrary probability of success is desired (i.e., a desired robustness level). This can be expressed as:

$$1-P\{\text{no extraction}\} = 1-10^{-s} \quad \text{Equation (10)}$$

Where s is selected, for convenience, to represent the exponent of target probability of no extractions. Using Equations 7 and 10, the following expression may be obtained:

$$s = \frac{X_j}{X_0} \frac{pE_k}{\ln(10)} \quad \text{Equation (11)}$$

This equation can be used to study the tradeoffs between the number of embedded watermarks E_k (which directly influences the transparency requirement of the system), the ratio involved in the extractor design, $r_j = (X_j/X_0)$, and the probability of success (i.e., $1-\text{Prob}\{\text{no extraction}\}$). Figures 13 and 14 show this relationship graphically. In Fig. 13, the exponent, s , is plotted against E_k , ranging from 1,000 to 40,000, for two limiting values of $p = 0.1$ and 1, and $r_j=10^{-3}$. For a 10^{-6} (or better) target probability of no extractions (i.e., for $s = 6$), this system requires E_k of about 140,000 (not shown on the plot) to ensure desired performance down to $p=0.1$. In Fig. 14, with r_j ratio of 10^{-2} , this target probability is reached when only about

$E_k=14,000$ distinct watermarks are embedded. The above example illustrates how the embedder key size can be traded off with desired security and extractor complexity.

ATTACK SPECIFIC SECURITY ENHANCEMENTS

Protection Against Blind Attacks

Performance of the present invention against some attacks described above warrants some attention. In the case of a blind attack, it is a well-accepted fact that it is always possible to add enough distortion to a watermarked content to render the embedded watermarks unextractable. The objective of a good attack is to put the right kind of distortion at the right place in order to disable watermarks with minimum damage to the host signal. The watermarking system of the present invention makes this task harder because of several security features that are incorporated into the system. Some of these features comprise:

- a. Multiple watermarking algorithms may be deployed with different robustness profiles. In order to disable all of them, an attacker must use a combination of attacks, which increases the damage to the host content.
- b. Not all potential insertion opportunities are used for embedding the watermarks within the content. If an attacker cannot find the exact location of the watermarks, he must attack all hiding places and thus introduce unnecessary and ineffective distortions.
- c. Experiments with an extractor with objective to find minimum necessary distortion that removes watermarks is made difficult because:
 - i. Extractors randomly select stego keys in each run, and several detection outcomes of the same content do not necessarily produce the same results.
 - ii. Different extractors have distinct stego key sets, so the outcome of experiments with one extractor is not necessarily the same as the outcome with another.
- d. Even if an attacker finds a successful attack on a particular content, through a sophisticated and lengthy procedure, e.g., many runs, many extractors, it is not useful for other content, where a different set of hiding places are used.

- e. Even if an attacker finds a successful attack on many contents, through a sophisticated and lengthy procedure, and distributes an attack algorithm against them, content owners may adjust embedding parameters (technology selection, watermark strength, etc.) for the next content release in order to thwart existing piracy systems.

The security and robustness of the watermarking system can be further enhanced through additional features of the present invention. These features will be described herein. As described above, any added advantage in one aspect of the watermarking system may be sometimes traded off for other desirable characteristics of the system. For example, an added advantage in watermark robustness may be traded off for reduced computational complexity of the extractors. The concepts that are disclosed herein are described in the context of improving security and robustness characteristics of the system but these advantages may be readily traded off to meet other specific system requirements as mandated by the application, customer requests or the success of circumvention attempts.

Reporting Delay and/or Avoidance

The main object of this security feature is to introduce certain amount of uncertainty in the reporting of watermark detections. To this end, the extractors may randomly discard some detections, or delay the reporting of a detection if the content is a function of time. For example, any particular detection may be discarded (i.e., not reported to the user) with a probability $p = 0.5$. This would effectively reduce the number of watermark detections by 50%, which is acceptable if the number of watermark detections per content is large. This way, an attacker cannot be certain about the success of his watermark removal attempt if a single run of the extractor produces no detections. In fact, for the above example and with a target probability of missed watermarks of 10^{-3} , the attacker needs, on average, to run the extractor at least ten times to correctly assess the result of his attack.

Alternatively, or in addition to the above probabilistic reporting technique, watermark detections may be reported on a time-shifted basis. One approach is to make reports at predefined time intervals. For example, instead of reporting the detection of every watermark separately, detections may be reported simultaneously at 5-minute intervals. Alternatively, the detected watermarks may be reported with a delay that is randomly assigned. For example, the

delay for reporting the detected watermarks can be between zero and five minutes with uniform distribution. In certain applications, where the multimedia content is screened for the presence of watermarks prior to being played and/or recorded (e.g., in TIVO applications or any other non-real-time application with considerable buffering), the watermark reporting may be advanced (as well as delayed) so that any reported watermark could correspond to a past, present or future segment of the multimedia content.

Enhanced Watermark Detection by Weight Accumulation Algorithm

A Weight Accumulation Algorithm is developed to combine detection of multiple damaged watermarks in order to make a successful watermark extraction. The algorithm may be best understood by assuming a watermark that is N bits long and comprises a predefined bit pattern. It is further assumed that k strings of bits per second are assembled and compared to a template; if the number of errors (i.e., mismatches between the assembled string and the template) is less than or equal to e , then a watermark is detected. In this case, the probability of false detections per second is:

$$P_f(N, e) = 1 - (1 - 2^{-N} \sum_{j=0}^e \binom{N}{j})^k \approx 2^{-N} k \sum_{j=0}^e \binom{N}{j} \quad \text{Equation (12)}$$

Equation (12) describes the relationship between probability of false detections, watermark packet lengths, number of attempts in extractor (i.e., computational efficiency of extraction) and error tolerance in template matching. If a zero error tolerance is specified (i.e., zero mismatches found), then the minimum packet length may be calculated as:

$$N_{\min} = 1 - \text{floor}\left(\frac{\log(P_f(N, 0) / k)}{\log(2)}\right) \quad \text{Equation (13)}$$

Obviously, in order to be able to detect watermark packets in the presence of noise and other impairments that may have damaged the embedded watermarks, it is advantageous to increase the number of errors that can be tolerated in each watermark packet while maintaining a desired false positive rate with optimum computational efficiency. Equation 13 may be used to study the tradeoffs between these characteristics and the watermark packet length. For example, for a target probability of false detections per second equal to 10^{-12} and for an extractor that performs 20 extraction attempts per second (with zero error tolerance), the minimum number of bits per watermark packet may be calculated to be 46. As the packet

length increases, so does the error tolerance, i.e. the number of errors per packet that is acceptable in extractor, as shown in Fig. 15. Alternatively, the ratio between the number of errors per packet and the packet length may be calculated to obtain the error tolerance in terms of percentage of bits that can be in error, as shown in Fig. 16.

As can be seen from Fig. 15 and Fig. 16, error tolerance may be improved by increasing the packet length. However, increasing the packet length makes the watermark larger, which will reduce the number of watermarks that can fit within a content. In addition, embedding longer packets may introduce additional artifacts within the content, may increase the processing requirements in the extractor, and may increase watermark vulnerability to synchronization attacks. An alternative option to using longer watermark packets is to cumulatively analyze multiple detected watermarks to achieve better error resiliency. One such technique used in the present invention is the Weight Accumulation Algorithm (WAA), disclosed below.

The details of WAA may be better illustrated by considering the detection of a watermark packet consisting of N bits that is contaminated with e_1 bit errors. The number of errors in a watermark packet may be determined by comparing bits of the detected packet to a pre-defined template of bits, which represents the error free watermark packet. Alternatively, if Error Correction Codes (ECC), such as Reed-Solomon codes, are used to form the watermark packets, ECC decoding of the watermark packets would produce an error count for the decoded ECC packets. Upon detection of such packet, the number of errors, e_1 , may be too large to declare a successful watermark extraction with high levels of confidence, but the detected watermark packet may still carry significant information about the presence of the watermark. The significance, or weight, of this information can be expressed as:

$$W(N, e_1) = -\log_{10} \left(2^{-N} \sum_{j=0}^{e_1} \binom{N}{j} \right) \quad \text{Equation (14)}$$

which corresponds to the logarithm of the probability of false detection of a watermark consisting of N bits with up to e_1 errors in a single attempt. Obviously, the weight of a detection increases as the number of errors decreases. Fig. 17 illustrates the relationship between the weight of each detection and the number of errors for a 100-bit long watermark packet, in accordance with Equation 14. The plot in Fig. 17 indicates that detected packets with lower error counts have a much larger weight than packets with high error counts.

Next, the conditional probability of detecting another watermark packet consisting of N bits, with up to e_2 errors, within a time interval of T seconds after the first watermark detection may be calculated. During T seconds, there will be kT extraction attempts, and the probability that at least one is successful is expressed as:

$$P\{(N, e_2) | (N, e_1)\} = 1 - (1 - 2^{-N} \sum_{j=0}^{e_2} \binom{N}{j})^{kT} \approx kT 2^{-N} \sum_{j=0}^{e_1} \binom{N}{j} \quad \text{Equation (15)}$$

The joint probability of detecting two watermarks with e_1 and e_2 errors, respectively, within the time interval T , is obtained as the product of probability of the first event and the conditional probability expressed by Equation 15. By applying the logarithm to the product, the total weight of the joint events can be expressed as:

$$W(N, e_1, e_2) = W(N, e_1) + W(N, e_2) - \log(kT) \quad \text{Equation (16)}$$

Extending this result to include a string of detections that occur at instances t_1, t_2, \dots, t_m , with respective error counts e_1, e_2, \dots, e_m , the total weight of this string of events is:

$$W(N, e_1, e_2, \dots, e_m) = \sum_{j=1}^m W(N, e_j) - \sum_{j=2}^m \log(k(t_j - t_{j-1})) \quad \text{Equation (17)}$$

If total accumulated weight of the string of watermarks reaches or exceeds a threshold value, it may be concluded that the watermark is present with sufficient confidence, i.e. with sufficiently low probability of false detection.

It should be noted that it may not be desirable to accumulate watermarks that have very low weights since it burdens the processor without contributing significantly to the weight accumulation. Further, it may not be desirable to accumulate watermarks separated by a large time interval since the approximation used in equation (15) is valid only if:

$$kT 2^{-N} \sum_{j=0}^{e_1} \binom{N}{j} \ll 1 \quad \text{Equation (18)}$$

For example, assuming $k = 20$ and $N = 100$, with a maximum acceptable number of errors per packet of 26 and maximum time interval between detections of 5 minutes, the left hand side of Equation 18 becomes 0.005, which satisfies the criterion specified by Equation 18.

In the above described WAA example, even watermarks with bit error rates as high as 26% are taken into consideration. Thus, accumulation of weights due to several successive or

properly spaced detections would produce reliable detections with a high degree of confidence even if individual watermark packets are highly damaged. Further refinements of the WAA algorithm may include utilizing the heart-beat or periodicity of the redundantly embedded watermarks to improve the reliability of detections. Since the same watermark packets may be embedded back-to-back (or with a predefined spacing) throughout the content, only watermark detections with correct separation may be included in the accumulation process. For example, it may be anticipated that the separation between valid watermarks should be a multiple of the duration of the watermarks (plus or minus some small fraction to allow for small deviations due to impairments). This way, the rate of false detections, due to randomly occurring or mis-aligned watermarks may be reduced.

The weight accumulation algorithm described above is particularly effective against blind pattern matching attacks, discussed above. The cutting and swapping algorithm may indeed damage watermarks so that none of them is individually recognizable. However, it will be much more difficult to sufficiently damage all watermarks so that even weight accumulation algorithm is not able to detect them. For example, assuming that the content segments that are swapped have a 50% chance of producing a bit error and further assuming that the system is designed to tolerate bit error rates of up to 26%, the swapping must be done over 52% of the content to bring the chances of individual watermark detections down to 50%. It is expected that swapping such a large percentage of the content would lead to serious degradations in content quality and significantly reduce the value of such content. It is also worth noting that while the above description of the WAA algorithm has been presented by describing multiple watermark packets that are separated in time domain, it is entirely possible to implement this technique for watermark packets that are separated in space, spatial frequency domain, temporal frequency domain or a combination of the above.

Additional countermeasures can also be developed to reduce the effectiveness of the blind pattern matching attack. For example, watermarks with variable bit rates may be embedded within the content. This way, swapping of segments may only disrupt the continuity of watermarks within a limited range of bit rates. For example, for a low bit rate segment of the content, the swapped segments may be much shorter than each bit interval and thus may not play a significant role in determining the value of the embedded bit. On the other hand, for a high bit rate segment of the content, the swapped segments may contain intact watermark

packets that are perfectly detectable. Using a system with mixed bit rate values forces the attacker to locate similar content segments of different durations and assess the success of his/her attack in a trial and error approach. This task, if possible, would be extremely time consuming and may never produce a content with acceptable quality.

Embedding with variable bit rates may be accomplished using inter-packet, intra-packet, or a combination of the two, bit rate variations. To illustrate these techniques, it is helpful to consider a watermark that is comprised of N bits. Each watermark packet is typically independently (and redundantly) embedded throughout the host content. In fixed bit rate watermarking protocols, all N bits of *every* watermark packet occupy the same length of time or spatial dimension within the host content. With the proposed intra-packet bit rate variation scheme, individual bits within each packet may occupy a different time or spatial real estate while maintaining a constant watermark packet rate throughout the content. Alternatively, in the inter-packet bit variation scheme, bit rates may be varied from one packet to another (i.e., variable packet rate) while keeping the bit rate within the packets constant. This way, some packets, as a whole, would potentially survive the swapping of segments and produce the desired detection results. This is in contrast to the intra-packet scheme, where survival of some bits may not be sufficient for the detection of the packet as a whole. In addition, in inter-packet bit variation scheme, each watermark packet maintains a constant duration (or extent), which could facilitate the detection of watermark boundaries. Further, the pattern of bit rate variation can be fixed for all embedded watermarks or may be varied in a pseudo-random fashion. The above concept is readily applicable to frequency domain or spatial domain watermark embedding schemes, as well. In such cases, the number of samples of the host signal that are used to carry watermark bits in the frequency or spatial domains may be varied in accordance with one or all of the above techniques.

Enhanced Watermark Detection by Time Diversity Decoding

Another technique that takes advantage of combining several weaker watermark detections is time-diversity decoding. Using this technique, 3 or more detected watermark packets with unacceptable number of errors, e_1, e_2, e_3, \dots , are collected and bit-averaged to produce a single watermark packet. Bit-averaging is achieved by counting the number of zero and one values for each bit position, and selecting an output bit value that corresponds to a

majority of input values. The output packet is again compared to the template and errors (mismatches) are counted. If the result is within an acceptable range of errors, successful detection is declared. The success of this technique is predicated on channel noise being a zero-mean, independent random variable so that after sufficient averaging, the true bit values are revealed. In order to ensure an unambiguous outcome, there needs to be an odd number of detected packets before such averaging can take place (that is, if detected packets contain binary symbol values). Nevertheless, if the averaging is performed when an even number of packets are present, and there are equal number of zeros or ones in a given bit position, either a zero or a one may be selected randomly as the detected bit value.

Methods and apparatus for enhancing the robustness of watermark extraction from digital host content using the WAA and time diversity decoding techniques discussed above may be implemented on the Extractor Apparatus 600 discussed in connection with Fig. 8. For example, in an exemplary embodiment the embedded host signal 560 containing the digital host content is received, e.g., at a receiver or other device incorporating a receiver (such as Extractor Reception Device 610 at Extractor Apparatus 600). The Watermark Extraction Device 610 extracts watermarks from the host content in accordance with a stego key (e.g., from the stego key selection device). In this embodiment, the Watermark Extraction Device 610 may be adapted to:

- (a) extract a first watermark;
- (b) assess a number of errors in the first extracted watermark; and if the number of errors is above a first pre-determined value;
- (c) extract at least a second watermark;
- (d) assess a number of errors in at least the second extracted watermark; and

If the number of errors in at least the second extracted watermark is above a second pre-determined value, the Watermark Extraction Device 610 may combine the extraction results of steps (a) and (c) to cumulatively assess the validity of the first extracted watermark and at least the second extracted watermark. Note that if number of errors in either the first or the second watermarks do not exceed the respective first and second pre-determined values, the combining of the extraction results does not occur and subsequent extractions and corresponding error assessments may take place in the same manner. Furthermore, the value of either the first or second pre-determined thresholds may remain unchanged throughout the

detection of content or may change dynamically in accordance with a deterministic or probabilistic (or pseudo-random) technique. Such variation of threshold values may vary the robustness or enhance the security of the system.

The combining may comprise assigning weights to the first and at least the second extracted watermarks. The assignment of the weights may be done by the Watermark Extraction Device 610 or a separate processor associated therewith (not shown). The weights assigned to the first and at least the second extracted watermarks may be added to obtain an accumulated weight value. The accumulated weight value may be compared to at least a first pre-defined reference value to assess the validity of the extracted watermarks. The assigning of the weights may be adapted in accordance with soft decision information. Such soft decision information may comprise probability values.

The first and at least the second extracted watermarks may be separated by a pre-determined interval. The pre-determined interval may be a function of the duration of the embedded watermarks. The separation may be a multiple of the duration of the embedded watermarks.

The Watermark Extraction Device 610 may combine the results of steps (a) and (c) only if the number of errors in at least the second extracted watermark does not exceed a third pre-determined value. The third pre-determined value may be selected in accordance with at least one of a desired robustness, computational efficiency or false positive rate of the extraction.

The number of errors in the first and at least the second extracted watermarks may be obtained by comparing the detected watermark symbols to a pre-defined template. Alternatively, the number of errors in the first and at least the second extracted watermarks may be estimated by decoding watermark packets that are Error-Correction-Code encoded.

The first and the second pre-determined values may be determined in accordance with at least one of a desired robustness, computational efficiency, or false positive rate of the extraction.

The Watermark Extraction Device 610 may extract at least a third detected embedded watermark prior to the combining. The number of errors in at least the third extracted watermark may be assessed. If the number of errors in the third extracted watermark is above a third pre-determined value, symbols of the first, second and at least the third extracted

watermarks may be averaged (e.g., at the Watermark Extraction Device 610 or a separate processor associated therewith) to produce a composite watermark packet. The number of errors in the composite watermark packet can be measured to assess the validity of embedded watermarks. The averaging may be adapted in accordance with soft decision information. The soft decision information may comprise probability values associated with individual bits of the first, second and at least the third extracted watermark.

Enhanced Watermark Detection by Soft Decision Decoding

Before describing the enhanced watermark detection via soft decision decoding, it should be noted that while watermark packets have been described as having 'bits', all concepts disclosed in the present invention are equally applicable to watermark packets with non-binary data symbols. The extension of the disclosed concepts of the present invention to the non-binary realm can be easily accomplished by persons of ordinary skills in the art.

In the description of previous decoding enhancement techniques, all detected bits have been represented as having either a '0' or a '1' value without considering any uncertainties in the detection of these bits. The exact method for detection of embedded bits varies from one watermarking technology to another. For example, in the Peak Amplitude Modulation technique described in U.S. Patent No. 5,828,325, quantized peak values of the host signal are compared to two sets of pre-defined values (one corresponding to embedded zeroes and the other set corresponding to embedded ones) and the pre-defined value closest to the quantized peak is selected to represent the detected bit. In other watermarking techniques, such as in typical Spread Spectrum detectors, the correlation value between the received signal and the carrier sequence is calculated and the sign of correlation peaks, if exceeding a pre-defined threshold, is mapped to detected ones or zeroes. In the above examples as well as in other detection techniques, in addition to specifying the detected bit values, it is also possible to provide a measure of certainty for the detected bit values. This additional information, sometimes referred to as soft information, would represent the likelihood of having detected the correct bit value. For example, in a spread spectrum detector with a detection threshold value of 100, two calculated correlation values of 101 and 5000 may be both decoded into the same binary value without differentiating between the two detections (this is known as hard decision decoding). In other words, the significance of the much larger correlation value is lost.

In the soft decision decoding technique of the present information, a probability value is assigned to each detected bit. In the context of above example, the first bit may be detected as having a binary value of say zero, with probability 0.55 while the second bit may be detected as having a binary value of zero, with probability 0.95. In order to illustrate how this may improve decoding of watermark packets, the example of a 100-bit packet with 26 errors may be revisited. In the absence of soft-decision information, an error count of 26 is produced, which may be too high to be considered a successful detection. In the presence of soft-decision information, the probable number of errors may be calculated as:

$$E = \sum_{j=1}^N f(p_j) \quad \text{Equation (19)}$$

Where N is the number of bits per watermark packet, p_j is the soft-decision probability value for the j^{th} bit and $f(p_j)$ is equal to p_j , if there is a mismatch with the reference template at j^{th} position, and is equal to $(1 - p_j)$, if there is an agreement with the reference template at j^{th} position. Returning to the above example, there are 74 bit positions that match the template values and 26 bit positions that do not. Assuming $p_j=0.9$ for all correctly matched bits and $p_j=0.6$ for all erroneous bits, Equation 19 produces an error count of 23. This value reflects a better assessment of the true number of errors in this packet than the previous count of 26 and can produce extraction results with higher reliability.

Equation 19 only represents one method of incorporating soft information using likelihood measures produced with bit level granularity. Alternative, or additional, techniques may be used to assign likelihood measures with packet level granularity. In spread spectrum systems, for example, the correlation value, depending on the specific implementation details of the system, could represent a detected bit, a group of bits, or a full watermark packet. In general, whenever the detection process involves comparison of some calculated value or function to one or more pre-defined values or functions, soft information representing likelihood measures may be generated and used to produce detections with higher reliability. Such techniques may also be applied to detect synchronization headers and calibration signals which may be present as embedded watermarks. These signals usually comprise fixed, re-occurring bit patterns that are embedded within the host content. Typical detection procedure involves comparing the pattern of extracted bits to the error-free synchronization pattern and assessing the presence of a synchronization signal based on the number of mismatches. This

procedure is similar to one described above in the context of watermark packet detection and thus can benefit from soft decision decoding.

The Weight Accumulation Algorithm and Time Diversity decoding techniques, described above, can also benefit from the inclusion of soft information. In these cases, incorporation of probability values in weight accumulation and averaging calculations of the two techniques would produce results that are more likely to represent the true state of embedded watermarks. In the case of the weight accumulation algorithm, soft decision probabilities may be used to produce new error counts (similar to Equation 19), which would then result in new accumulated weight values that are calculated by Equation 16. In the case of Time Diversity decoding, probabilities associated with each bit, as well as the value of the bit, may be averaged over several detected packets to produce a single packet with one set of probability values associated with each bit of the packet. Alternatively, the probability values associated with each bit may be examined prior to the averaging process in order to exclude individual bits (or the collection of all N bits that make up the packet) from the averaging procedure. This way, marginal bits and/or packets that are not detected with high degree of uncertainty (e.g., $0.5 < p < 0.65$) may be excluded from the averaging process. Incorporation of soft information into Time Diversity decoding further provides for the decision making to occur in the presence of either an odd or an even number of packets. In such cases, it is still possible to obtain an ambiguous outcome (i.e., when p is exactly equal to 0.5) but this outcome has a very small chance of occurring for either an odd or an even number of packets.

The assignment of probability values to various detections is greatly dependent on the watermarking technology and its parameters as well as the desired levels of system robustness and false positive rates. Fig. 18 exemplifies a set of probability values which may be used in a detection technique that relies on calculating correlation coefficient values for detecting the embedded watermark bits. In order to strike the right balance between the various system requirements, the assignment of soft decision likelihood measures and the various threshold settings in the watermark extraction system often requires experimental fine-tuning and verification.

Soft decision information generated according to the above techniques may also be used to improve the extraction of watermarks that use Error Correction Coding (ECC). In these systems, watermark packets are typically ECC encoded prior to their insertion into the host

content. Upon detection of embedded bits, ECC packets are assembled and decoded to produce error-corrected versions of the watermark bits. The use of soft decision information for improved decoding of ECC codes (e.g., BCH and Reed-Solomon Codes) is well known to those skilled in the art of signal processing. Soft decision decoding is especially beneficial if interleaved or product codes are utilized. In these cases, iterative decoding in conjunction with soft decision decoding provides superior error correction capabilities.

Methods and apparatus for enhancing the robustness of watermark extraction from digital host content using the soft decision decoding techniques discussed above may be implemented on the Extractor Apparatus 600 discussed in connection with Fig. 8. For example, in an exemplary embodiment, the embedded host signal 560 containing the digital host content is received, e.g., at a receiver or other device incorporating a receiver (such as Extractor Reception Device 610 at Extractor Apparatus 600). The Watermark Extraction Device 610 extracts watermarks from the host content in accordance with a stego key (e.g., from the stego key selection device). In this embodiment, the Watermark Extraction Device 610 may be adapted to:

- (a) decode individual symbols of an embedded watermark in accordance with at least one watermark detection algorithm;
- (b) assign discrete symbol values to the decoded symbols;
- (c) generate likelihood measures associated with the decoded symbols; and
- (d) assess the validity of the extracted watermarks in accordance with the discrete symbol values and likelihood measures.

The likelihood measures may comprise probability values.

The Watermark Extraction Device 610 (or a separate processor associated therewith (not shown)) may assess the validity of the extracted watermarks by multiplying each discrete symbol value by the likelihood measure corresponding to the symbol value to produce weighted watermark symbols. The weighted watermark symbols may be arranged in a pre-defined order to form a weighted watermark packet. The number of errors in the weighted watermark packet may be compared to a pre-determined reference value in order to assess the validity of the watermark.

The likelihood measures may be obtained in accordance with a set of pre-defined threshold values. Further, the likelihood measures may be generated in accordance with a distance between the decoded symbols and the threshold values.

The detection algorithm may comprise at least one of spread spectrum, autocorrelation modulation, peak amplitude modulation, or replica modulation techniques. The detection algorithms may be stored at the Watermark Extraction Device 610 or a separate storage device associated therewith (not shown).

Subsequent to the receiving of host content, at least one boundary of the embedded watermarks may be estimated by detecting a presence of a synchronization pattern embedded in the host content. The detecting of the synchronization pattern may occur at the Watermark Extraction Device 610 and comprise:

- (a) decoding individual bits of an embedded pattern in accordance with at least one watermark detection algorithm;
- (b) assigning discrete bit values to decoded bits of the embedded pattern;
- (c) generating likelihood measures associated with extracted bits of the embedded pattern; and
- (d) assessing a presence of the synchronization pattern in accordance with the bit values and associated likelihood measures.

Protection Against Differential Attacks and Analysis

Differential attacks described above are not very effective against the watermark embedding scheme of the present invention, simply because different contents have watermarks hidden in different places. Therefore, copying the transfer function from one original and watermarked content pair and applying inverse transfer function to another watermarked content would generate artifacts in the wrong places with little chance of affecting the watermarks themselves.

The content dependent stego key set is also an obstacle for an effective differential analysis, i.e. analysis of one difference signal may not be relevant for another content. However, a sophisticated attacker may combine results of many differential analysis attempts (from many different content pairs) in order to generate more or less complete picture of the embedder stego key set. Thus, in the watermarking system of the present invention, novel

masking techniques are utilized to conceal the presence of the watermarks and/or the stego key space corresponding to the embedded watermarks.

Differential analysis relies on the difference signal between the watermarked and unwatermarked versions of the host content to uncover the stego key. So, one goal of the watermarking system of the present invention is to render the difference signal as ambiguous as possible. This is sometimes referred to as masking the watermark signal. Masking can be done in several ways. One method is to process the watermark signal in such a way that despite the recovery of the proper difference signal, it may not be intelligible to the attacker. This method is described in the flow diagram of Fig. 19. After content acquisition (step 101) and generation of the watermark (step 102) in normal way, the generated watermark undergoes a masking procedure (Step 103) before being applied to the host content (step 104). Watermark masking may comprise encryption of the digital watermark, scrambling of the digital watermark or linear or non-linear processing of the watermark signal, etc. An attacker, having obtained the difference signal is not able to readily decipher and interpret the true meaning of the watermark signal.

A second method of masking manipulates the host content prior to the embedding of the watermark. This method is shown in the flow diagram of Fig. 20. After content acquisition (step 201), the content is manipulated (step 202) before generation of the watermark (step 203) and application of the watermark to the host content (step 204). In this case, the difference signal is not the true representation of the watermark since the generated watermark corresponds to the manipulated version of the host content and not the host content itself. The key consideration here is to design the manipulation technique so that the perceptual quality of the host signal is maintained. Another consideration is the security of the manipulation technique. That is, the extent and details of manipulation should not be easily discerned from the analysis of the signals. The latter may be achieved by changing the nature and/or parameters of the masking technique from one embedding to the next. By the way of example and not by limitation, such manipulation techniques may comprise phase distortion, linear or non-linear distortion or non-uniform resampling of the content.

A third method of masking is shown in Fig. 21. After content acquisition (step 301), watermark generation (step 302), and application of the watermark to the host content (step 303), the content is then manipulated (step 304) in order to mask the watermark. This

technique is similar to the technique described in Fig. 20 above, except in this case, masking is performed after the application of the generated watermark. It is generally assumed that watermarks themselves are immune to the masking transformation. It is further possible to combine any one of the systems described in Figures 19-21 to produce a system with two or more levels of watermark masking. However, in any masking configuration, two basic requirements must be met. First, the perceptual quality of the composite signal must be within acceptable limits, and second, specifics of the watermark signal must not be easily discerned from the analysis of the difference and/or the composite signal.

Aside from masking techniques, the watermarking system of the present invention renders differential analysis and/or attacks ineffective by incorporating different stego keys for each embedded content. Thus, if an attacker is successful in deciphering the stego key from one embedded content, he/she will not be able to use the recovered stego key to affect any other content.

Protection Against Collusion Attacks

Collusion attacks through differential analysis is not effective against the present invention, for the same reasons described above, providing that each embedding generates a distinct masking pattern. Further, the masking techniques described above are also effective against averaging and cut-and-splice collusion attacks. For a system that uses one of the above mentioned masking techniques, averaging and cut-and-splice attacks may only weaken the embedded watermarks, but joint extraction of multiple watermarks should result in eventual watermark extraction, as described above in the context of the Weight Accumulation Algorithm, Time Diversity decoding, and soft decision decoding.

Protection Against Oracle Attacks

Oracle attacks are generally complex, and not very effective against watermarking techniques that are signal dependent. This attack is further hindered by the uncertainty in the extractor response, i.e. no watermark extractions in a run does not necessarily mean that no watermarks were detected in the content. Other features of the present invention, such as variation of embedding algorithms, sparse embedding and joint extraction of multiple watermarks should also contribute to ineffectiveness of oracle attacks.

Protection Against Overwriting Attacks

Overwriting attacks are not effective against the present invention simply because different embedders use different stego keys for different embedding sessions. So, the extractor should have no trouble finding all watermarks even if embedded in different sessions. Reaction to the extraction of multiple watermark states is the responsibility of the application layer and may be dictated by the content owners. For example, one may decide to only react to the most restrictive watermark if multiple watermark states are extracted within the same content.

In order to prevent the denial of service attacks, described above, the watermark embedding devices of the present invention also include watermark extractors that examine the content prior to embedding. If pre-existing watermarks are detected within the content, this information is conveyed to the embedder and possibly to the application layer. Depending on the value/state of the pre-existing watermarks and the current value/state of watermarks, the embedder may decide to continue embedding or abort the entire procedure. Alternatively, or additionally, the embedding device may alert the user and/or the (legitimate) content owner regarding the discovered discrepancy. Identifying the legitimate content owner may require connectivity of the deployed embedding devices to a central database. Other safeguards could include requiring passwords, access cards or use of biometric information for enabling an embedding session.

Another method for preventing fraudulent access to the multimedia content through such overwriting attacks is to embed additional watermarks (as an independent layer) that contain ownership-related or any other additional information regarding the multimedia content. This additional layer may serve to provide a second level of authentication for the embedded content. A typical usage scenario for such system may be described as follows. A content owner embeds a multimedia content with one type of copy control watermark and an additional layer of watermarks that convey ownership information. The latter may be in the form of a serial number, which may serve as an index to a remote database. The content owner additionally registers his/her content as having one type of copy control state (i.e., the same copy control watermark that was embedded) and all this information is stored at a secure database.

In the extractor device, one of three actions may take place. First, the extractor may extract the copy control watermark only, and react according to the set of rules associated with that copy control state. Second, the extractor may only extract the second layer of watermarks containing ownership information, access the remote database of information to determine the copy control state and act according to the set of rules associated with that copy control state. Or third, the extractor may extract both watermark layers, access the remote database to ascertain copy control state information and verify it against the copy control state obtained from extraction of the watermarks. In case of a discrepancy, the extractor may decide to notify the owners, select the most restrictive copy control state, trust the information obtained from the database, etc. It should be noted that embedding of a second layer of watermarks requires re-assessment of robustness, security, transparency and computational complexity of each individual watermark layer and the system as a whole. In addition, payload requirements of the new watermarks would most likely change since a modest number of additional bits (e.g., 30 to 40 bits) may be necessary to carry the desired information within the watermark or act as a pointer to a remote location.

The above described technique provides multiple methods for checking the validity of extracted watermarks, all made possible by including two different types of information. It is similarly possible to extent this method to include three or more different layers of watermarks to provide additional protection. An attacker now has to overwrite all layers of watermarks in order to claim success. These attempts may further be discouraged by requiring all users to provide identity credentials prior to each embedding. These credentials may be verified or authenticated, and in the presence of pre-existing watermarks, verified against the credentials of prior owners. The latter credentials may be carried within the pre-existing watermarks, or acquired by accessing a remote database in accordance with the extracted information from the watermarks. In case of discrepancies between the two credentials, embedding may be disabled or both owners may be contacted to resolve the issue.

Protection Against Embedder Based Analysis Attacks

Differential analysis based on test signals is more dangerous than differential analysis based on a common content, and reliance on watermark masking techniques may not be sufficient against these attacks. However, these attacks may be thwarted by implementing test signal extraction and embedding avoidance techniques described below.

Masking techniques described above are also effective against embedder-based attacks. Additionally, the watermarking system may employ embedding prevention techniques to disable watermark embedding when input signals with certain properties are extracted. This procedure is described in the flow diagram of Fig. 22. Following the Content Acquisition Step (401), the host content is analyzed in Content Analysis Step (402). Based on the result of this analysis, it is determined whether or not the input content contains a prohibitive condition at a Decision Step (403). If a prohibitive condition is discovered, then the embedding of watermarks is disabled in Watermark Generation Disabling Step (404), otherwise, normal procedures for generation and application of the watermarks are carried out in Watermark Generation Step (405). Alternatively, if the presence of such prohibitive conditions are detected, the embedder may generate a false (e.g., dummy) signal instead of the legitimate watermark signal. This technique serves to produce even more misleading results in the presence of such attacks.

An exemplary list of signals that could affect watermark generation are impulse signals, sinusoidal signals, flat image signals, edge image signals, step functions, signals with specific temporal or frequency characteristics and other custom-designed signals. These signals may be recognized in real-time by content recognition techniques. For example, the incoming signal, or its attributes, may be compared to stored signal patterns, or their attributes, residing in a memory location. Alternatively, the attributes of the incoming signal may be calculated on the fly and compared to stored versions of generated reference patterns. To illustrate further, in audio applications, an impulse response may be recognized by measuring the peak-to-average value of the incoming signal over a finite time period and comparing it to a set of reference ratios. Other waveforms may be recognized by comparing one or more of their inherent or calculated characteristics, such as their energy or correlation to reference functions and/or values stored in a look-up table. Opting for the look-up table implementation allows for periodic update of the stored waveforms and/or prohibitive conditions.

Implementation of such embedding prevention techniques reduces the overall efficiency of embedding. That is, since some of the prohibited signals may naturally occur in multimedia content, some portions of the host content may not be embedded and thus the robustness of the embedded watermarks may be reduced. By the same token, due to the fewer number of embeddings, transparency of embedded watermarks may improve. In designing

watermarking system of the present invention, the number and types of prohibited signals may be adjusted, from one application to the next or from one embedding to the next, in order to fulfill the required security, robustness and transparency of the embedded watermarks.

Protection Against Protocol Attacks

Protocol attacks do not affect watermarks themselves, but still can render watermarking system ineffective. As discussed above, protocol attacks can be classified as internal and external. Internal attacks are concerned with information flow within the device, while external attacks are concerned with signal manipulation outside the device.

Internal attacks are designed to interrupt information flow within the device in order to gain access to information or to modify it. Therefore, it is essential to insist that all information flow within the device is cryptographically secure. This means that no information should be exchanged in the form of 'plain text'. Different software modules, such as extractor and application level software, should use cryptographic authentication techniques in order to prevent manipulation of the data. Those techniques are well established within those skilled in the art, and they are not subject of this disclosure.

External attacks may comprise scrambling and descrambling operations, as previously described. It may be possible to automatically detect unauthorized scrambling of the content and abort recording or playing of the content, generate warning signals, and/or notify authorized personal. This task may require analyzing certain characteristics or statistical properties of the content in order to discern whether or not they conform to the true characteristics or statistical norms of the typical content. For example, scrambling an audio signal "whitens" the frequency spectrum of the content. Detection of this condition in an input signal may trigger an extractor to generate a warning signal or initiate a restrictive action. In some ways, this technique resembles the embedding prevention techniques described above, where the incoming signals are analyzed to determine whether or not they contain special characteristics. Analogously, implementation techniques for storage and analysis of reference signals/conditions in the embedder apparatus are equally applicable to the extractor apparatus. Thus, the extractor may also look for the presence of special test signals and adapt the extraction process based on the presence of such test inputs (e.g., extraction prevention). The

main goal of signal analysis may also be to recognize signal features indicative of potential scrambling.

Alternatively, in order to circumvent these types of attacks, additional signal processing steps may be carried out at the output of the device that is imperceptible for normal usage, but interferes with the descrambling operation. These signal processing steps may be considered as *almost* lossless signal transformations ('almost' being the key word) that, in the absence of malicious signal conversions, produce imperceptible signal degradations, but in the presence of such attacks they significantly damage the perceptual quality of the host signal. For example, the success of the signal conversion technique, described by Equations 1 and 2, and illustrated in Fig. 1, is only possible if the descrambler receives substantially the same bit stream as the one leaving the scrambler. In other words, in the absence of distortions in the "channel" depicted in Fig. 1, the descrambling operation produces an output bit stream, b'_n , that is identical to the input bit stream, b_n . However, in the presence of channel distortions, the output bit stream may look drastically different from the original bit stream. Such channel distortions may be intentionally introduced as an additional signal transformation step, such as a simple D/A followed by A/D conversion, somewhere between the scrambler output and descrambler input in the extraction process. Other substantially imperceptible processing steps include resampling, slight nonlinear distortion or all-pass filtering (in the case of audio signals). In the case of audio signals, an especially effective technique is low pass filtering with a high cutoff frequency, e.g. above 20 kHz. This is a substantially imperceptible modification of audio signals since human ear sensitivity decreases significantly with increase of audio frequency, and most of the energy of typical audio signals is concentrated around the lower frequencies. On the other hand, scrambled audio signals typically have a flat spectrum and low pass filtering of such scrambled content could remove a sizeable portion of the signal energy that, upon descrambling of the content, could produce an audio signal with significantly degraded quality. It is further possible to perform the above mentioned processing steps intermittently (e.g., randomly) or with different levels of variability. For example, the cutoff frequency of the audio filter in above example may be adjusted in a 2 KHz range around the original 20 KHz value. These variations may degrade the multimedia content to various degrees (or in some instances, not at all) but they serve to frustrate an attacker by producing inconsistent results.

There are also scrambling techniques that are not bit sensitive and can be executed in analog domain. Most of these techniques are based on signal modulation/demodulation type of attacks, as described above. Those attacks are much more limited in the number of variations as compared to digital scrambling attacks. Therefore, it is possible to anticipate some of them and prepare countermeasures in the extractors. One type of countermeasure is to insert specific descramblers prior to the regular search for watermarks. For example, it is possible to invert every other sample (which is equivalent to multiplication with $\cos(\pi f_s t)$, with f_s being the sampling rate) and then conduct a search for embedded watermarks. Thus, in addition to searching the input signal in the normal way, one or more such descramblers may be utilized to examine modified versions of the input signal. Alternatively, in order to limit the computational cost of extractors, an extractor may examine the input stream in a normal way and only occasionally turn one or more such descramblers on to check for possible transformations.

Another strategy for withstanding such attacks is to deploy watermarking techniques that are invariant to typical analog scrambling techniques. For example, some distributed feature quantization techniques compare energies between two non-overlapping time intervals. This relationship is typically unaffected if the multiplier function has a period that is much shorter than the considered intervals. Other techniques include using an embedder to insert additional watermarks that are only detectable in the converted domain. For example, if an attacker develops and distributes a scramble/descramble attack that is based on spectral inversion of audio signals, such that the zero frequency component is moved to 24 kHz and the 24 kHz frequency component is moved to zero, and if the extractor searches for watermarks in the frequency band between 500 and 4000 Hz, then the embedded watermarks in the next release of the content may be inserted in the frequency range between 20,000 and 23,500 Hz. This way, the scrambler box would invert the audio spectrum and bring the watermarks from 20 to 23.5 kHz range down to 0.5 to 4 kHz range where extractor would find them. It is not necessary to hide all watermarks in the converted domain as it only suffices to produce a few detections in order to render the existing attack ineffective. In other words, it may not be necessary to burden the extractors with additional processing loads; instead, by anticipating (or knowing) the actual scrambling attacks deployed by the pirates, watermark

embedding process may be modified for insertion of additional watermarks that are tailored to be detected only under specific transformations.

Also, it should be noted that the introduction of additional signal processing operations between the scrambler and the descrambler may prevent analog scrambling attacks, as well. For example, an attack that relies on modulation and demodulation of audio signals with a frequency equal or close to f_h , the highest possible frequency in the analog signal, can be made ineffective by inserting a low-pass filter in between the two operations. Note that modulation of an audio signal with a carrier at frequency f_h , actually inverts the spectrum, so that the energy of the modulated signal is concentrated at high end of the audio spectrum, but a considerable amount of the signal may be removed if this inverted spectrum is subjected to low-pass filtering. After descrambling (i.e., demodulation), the spectrum is reversed again, but most significant, low frequency parts would be missing at the output.

Forensic Tracking

The main objective of forensic tracking (also known as transaction coding, fingerprinting, traitor tracing, copy tracing, etc.) is to embed distinct watermarks into each copy of the multimedia content in order to trace the origins and distribution paths of the pirated content and identify the participants in the piracy chain. As previously described, the forensic tracking information may be embedded as a separate watermark layer with a potentially high payload capacity. This method involves embedding each copy of the multimedia content with digital watermarks that carry identification information. Upon recovery of a pirated content, the origin of the content is revealed by extracting the identification information contained within the embedded watermarks.

Additionally, or alternatively, any set of watermarks that are embedded in accordance with the present invention may inherently carry forensic information. Thus it may not be necessary to embedded an additional layer of watermarks solely for the purpose of forensic tracking of the origin of a content. This may be accomplished by assigning a unique set of embedding stego keys to each embedding device. Upon recovery of a content under suspicion and extraction of the embedded watermarks, the pattern of embedding opportunities would identify the culprit device. In other words, each set of embedding stego keys may serve as a serial number for one embedding device. A similar method involves utilization of masking

parameters as serial numbers. Different masking techniques were previously illustrated in Figs. 19-21. In order to enable forensic tracking of the content, it suffices to assign a particular masking pattern (or masking parameters) to each embedding device. Upon recovery of a suspect content, the parameters of the masking process may be identified and traced back to the embedding device. For a masking technique with a pseudo-random phase generator, for example, this “serial number” may comprise an initial seed value that is assigned to each embedding device.

Forensic tracking of the multimedia content in the present invention is not limited to the identification of the embedding device. In fact, each embedding of the multimedia content may be identified by a unique embedding stego key and/or masking parameter. In this case, a new set of embedding/masking stego keys are issued every time a new embedding session is started. Obviously, accurate accounting of embedding/masking stego keys are required in order to keep track of the embedded content. This may be accomplished, for example, by storing pertinent information in a secure database upon completion of each embedding.

Weak Detections

In achieving the proper balance between computational complexity, improving the detection robustness, and limiting the number of false detections, one or more threshold values are usually established for the detection of embedded watermarks. An example of such a threshold was previously disclosed as the number of erroneous symbols in an extracted watermark packet. Another example may comprise a threshold value for comparing the correlation between a detected feature of the input content and a pre-defined pattern. Specifically, in a watermarking system that hides the data in least significant bits of the content samples, extraction of watermarks may be conducted by correlating the least significant bits with a known pattern, and comparing the result to a threshold value. In these and other examples, relaxing the threshold values may produce more detections, but it also increases the probability of false detections.

One method of improving the system performance is to define two (or more) different threshold values. The first set of threshold values produces the so-called ‘strong’ watermarks that meet the desired false positive requirement. The second set of threshold values produces the so-called ‘weak’ watermarks that do not meet the required probability of false detections

but may trigger further actions. This concept was previously disclosed in the context of Weight Accumulation Algorithm and Time Diversity Decoding discussed above, where two or more weak detections were combined to potentially produce a strong watermark. In the present case, the detection of one or more weak watermarks (and no strong watermarks) can trigger subsequent actions that comprise adapting the extractor configuration, extending the detection interval of extractor operation (e.g., in time/space/frequency domains), enabling more powerful error correction techniques such as erasure correction, iterative decoding and soft decision decoding, or triggering a forensic-like analysis.

Modifying the extractor configuration: As discussed earlier under the “Stego Key Design” heading, an extractor that normally uses a given set of stego keys for its operation also has several additional sets of stego keys at its disposal for future use. In addition, the extraction stego key may vary from one extraction trial to the next, allowing the extractor to vary and tune its stego key parameters, as necessary. Accordingly, in reaction to the detection of weak watermarks, the extractor may use this flexibility in the usage of its stego keys to re-focus the search for watermarks, expand the extent of search locations, or even conduct searches outside possible embedding locations in anticipation of channel distortions. One way to accomplish these tasks is to increase the granularity of the overall search, particularly in portions of the extraction space that yielded the weak watermark. For example, the search may be re-focused to examine the particular audio channel, frequency range, or time scaling factor that produced the weak watermark. Increasing the search granularity may include using a faster clock rate to collect or analyze the host content samples, perform fine (as opposed to coarse) synchronization, or perform mathematical operations with a higher granularity and precision. Another example includes switching to a new stego key that instead of searching a particular time-scaling space with, for example, 5% granularity, it searches the same space with 1% granularity. Alternatively or additionally, it is possible to enable additional watermark detection algorithms that were not part of the original extraction stego key and expand the search to include additional time slots, frequency bands or pixel locations. In order to keep the computational costs within a reasonable limit, all or some of the above operations may be discontinued once a strong watermark is detected, or if no watermarks are detected within a given time period.

Extension of extractor operation: The extension of extractor operation is applicable to both continuous and intermittent modes of operation. Further details of the intermittent detection mode will be described shortly. But it suffices to know that the extractor in intermittent mode only examines selected portions of the host content, and it becomes inactive if no watermarks are detected from the examined portions. In accordance with an embodiment of the present invention, the extractor activity may be extended if the extractor fails to detect a strong watermark but it detects one or more weak watermarks from the host content. This way, the extractor has the opportunity to examine additional regions of the host content, which may lead to the detection of strong watermarks. The frequency, duration or range of such extensions is a system design parameter that is determined in accordance with the desired system robustness and available computational resources. The extension of detection operation must be interpreted according to the type of host signal. For example, in a still image, the extension of detection operation may comprise examining additional pixel areas within the still image that were not previously examined. In case of an audio signal, additional time intervals, audio channels, or both may be examined. It is also possible to extend the detection operation to examine additional spatial or temporal frequency ranges within the original detection interval or region. The extension of extractor operation can further be combined with any of the techniques described under in the context of modifying the extractor configuration.

Enabling more powerful error correction techniques: Weak watermarks could also trigger more aggressive error correction strategies that may have been dormant during the initial screening of watermarks. Examples of these techniques include time diversity decoding, erasure correction, soft decision decoding or iterative decoding of watermark packets. Time diversity and soft decision decoding were described earlier. Eraser correction and iterative decoding are known terms of art in the area of error correction techniques, the application of which depends on the particular error correction codes used in the watermarking system.

Forensic analysis: The presence of weak watermarks could also start a forensic detection process. When conducting a forensic detection analysis, some knowledge of the original host signal or the embedded watermarks is usually required. In some systems, the original unmarked host signal is available to the extractor and may be used to perform a non-blind detection. In such cases, the original unmarked content may be used as a reference to undo the distortions that may be present in the embedded content. By the way of example and

not limitation, this process may comprise aligning the salient points (i.e., prominent features) of the original and received content in order to undo any time/space/magnitude scaling of the received content that may have prevented the detection of strong watermarks.

Other techniques take advantage of prior knowledge regarding the range of embedded watermark values. This information can help the extractor in two ways. First, the detected watermark values that are outside this range can be automatically discarded from consideration. Second, since the extractor is only required to search a subset of all embedded values, it can relax its error tolerance threshold without incurring any penalty in the probability of false detections. The effectiveness of this approach is highly dependant on the structure of data within the watermark packets and the availability of external information to the extractor. For example, if a watermark packet contains a “date-of-embedding” field, then the extractor can safely discard all detections that indicate a future date of embedding. Similarly in an example where a watermark structure contains a “type-of-media” field, the extractor, when processing an audio signal, can eliminate all watermark candidates that correspond to still images or video signals.

It should be noted that while the above described reactions to weak watermarks have been described, for clarity, under separate categories, there is some degree of overlap between the described categories, and an ‘application’ layer may decide to employ all or some of the described operations. Additionally, all such operations are applicable to extractors operating in continuous or intermittent modes of detection.

Intermittent Detection

The lack of adequate computational resources may sometimes necessitate intermittent (as opposed to continuous) operation of the extractor. This may be the case in consumer electronic devices, mobile phones, hand-held media players, and the like, where the implementation of a continually operating watermark extractor simply consumes too many computational resources. Even in environments with more computational resources, it may be desirable to allocate the excess computational capability to other high priority applications. In addition, the processing load associated with the watermark extraction pre-processing steps (and not the extraction process itself) may place an undue burden on the system resources. For example, screening a multimedia content for the presence of audio watermarks may require

such additional steps as decryption, de-interleaving, decompression or resampling of the content before the audio signal can be presented to the extractor. If in such systems, a content is merely being copied, transferred or uploaded from one device to another, performing such complex operations would unreasonably delay the transfer or upload of the content. Besides, it may not be justified to continually search a long multimedia content that is not likely to contain any watermarks.

For these reasons, it is sometimes preferable to activate the watermark extraction process for only limited periods of time. The duration and separation of extraction attempts may depend on the availability of computational resources, the particular watermark embedding and detection algorithms, the value of multimedia content, the type of multimedia content, and the nature of application. Obviously, the duration of each detection interval must be long enough so that a reliable assessment regarding the presence or absence of embedded watermarks can be made. In a typical scenario, when no watermarks are detected, the extractor becomes dormant until it is prompted to conduct another search. The time between each search is once again dictated by the available computational resources and the value of the content. Randomly spaced detection intervals work well for most applications since they provide added security by creating different detection results for each extractor run.

System Reactions to Copy Management Watermarks

In a copy management system, the detection of embedded watermarks can invoke different types of system reactions. The severity and duration of these reactions may depend on several factors, including but not limited to, the value (or state), type, spacing, or density of extracted watermarks. Furthermore, in some systems, the mere presence or absence of watermarks may also invoke a system reaction. In accordance with the embodiments of the present invention, system reactions may be classified as one of the following:

- 1) *Permissive Reactions*: This is expected to occur when the intended system operation conforms to the prescribed rules associated with the detected watermarks. For example, the detection of a “no-copy-allowed” watermark in a playback device should not affect the playback operation in any way. Other permissive reactions may occur when the detection of a certain watermark state initiates a grace period before a more prohibitive action is commenced. For example, the detection of a prohibitive watermark state may

initiate a warning signal to alert the user of the impending action, which may occur in the near future. In some applications, it may be desirable to disallow normal operation of the multi-media system in the absence of embedded watermarks. In such cases, the presence of extracted watermarks would be a necessary condition for continued normal operation of the system. Thus it may be possible to examine the type, value, density or spacing of the extracted watermarks to assess if the normal system operation should continue, or one or more of the restrictive reactions should take place.

- 2) *Conditional Permissive Reactions:* As opposed to unconditional permissive reactions, the detection of certain watermark states may trigger an intermediate reaction to determine whether or not the user would be allowed to access the content as he/she desires. Such reactions may require a response from the user. For example, the user may be asked to fill-out a form, view an advertisement, enter a password or pay a fee before he/she is allowed to access the content. Other conditional reactions may involve third party participation, comprising receiving a response (e.g., authorization, verification, etc.) from an actual third party, an automatic response from a remote entity (such as a server or a website), or an inquiry to a local database/file located at the user premises. In the event that the condition in questions is not satisfied, any of the remaining reactions described in this section may be commenced.
- 3) *Prohibitive Reactions:* These types of reactions normally occur when the intended system operation does not conform to the prescribed rules associated with the detected watermarks. For example, the detection of a “no-copies-allowed” watermark during a record operation may completely halt that operation, and optionally display a warning as to the reason for such prohibitive action. Other examples of prohibitive actions include, but are not limited to, muting and/or blanking the multimedia output, stopping the transfer/recording/playback of the content, and ejecting the medium that stores the content.
- 4) *Status Modification Reactions:* Status modification reactions can occur due to a conflict between the rules associated with the detected watermarks and the intended use of the content. In such cases, the multimedia content may be reversibly obscured (with an optional warning notice). Obscuration may also be preferred in systems where it is not feasible to produce a prohibitive reaction or its is desired to produce a high level of

annoyance to serve as deterrence for unauthorized usage or piracy of the multimedia content. Examples of such obscuration techniques include scrambling and encryption of the content. Since these techniques are generally reversible, they can alert the user to obtain a new authorization, upon perhaps a fee payment, for continued usage of the multimedia content. Other status modification reactions may occur in accordance with the prescribed rules of the detected watermarks. For example, in a 'generational' copy management system, the presence of a watermark state may trigger a remarking process that modifies or (over-writes) the embedded watermarks to represent a new watermark state. This remarking operation, for example, may involve overwriting a 'one-copy-allowed' watermark with a 'no-copy allowed' watermark while allowing the user to make a single copy of the newly modified content. Status modification may also be accomplished by changing the credentials associated with the host content by means other than watermarking. For example, a change in content status may be effected by changing the meta data fields associated with the content (e.g., header information), or changing the encryption status of the digital host content. Further examples of techniques in this category include compressing the content using proprietary algorithms or modifying the content (or attributes of the content) in such a way that it is only accessible by special software or hardware devices. In general, status modification involves the application of additional security features that render the content unusable. However, the user may be able to undo any such modifications by taking further actions.

- 5) *Degrading Reactions:* As an alternative to the above reactions, the perceptual quality of the multimedia content may be degraded while allowing the intended operation to take place. Some examples of various degradation techniques comprise re-sampling or down-resolution of the content, perceptual (i.e., lossy) compression, dynamic range reduction, spectral shaping (e.g., blocking, attenuating or supplementing certain spectral ranges of the content), addition of wow and flutter, phase distortion, intermittent blanking or muting the output, or partial scrambling/encrypting of the content to make is degraded but still recognizable (see, for example, commonly assigned U.S. Patent No. 6,889,943). One advantage of applying these techniques is that the degree and duration of degradation can be adjusted based on factors such as the

value and frequency of the detected watermarks and the importance/value of the content. Similar to the status modification techniques described above, some degradation techniques, such as partial scrambling or encryption, may be reversible while others, such as lossy compression, resampling, or dynamic range reduction, may not be reversible.

The duration of the various types of reactions described above could also be varied depending on the application and the type of detected watermark state. Thus while the detection of a “No Home Use” watermark may result in the permanent stoppage of play and ejection of the storage medium, the detection of a “No Internet Distribution” watermark may result in muting/blanking of the output signal for a short duration. The duration of enforcement may further be extended if additional restrictive watermarks are detected. The enforcement duration (and any further extensions thereof) may also be randomly varied in order to protect the system from analysis attacks.

The above described system reactions are not exclusive to the playback or recording devices, and are equally applicable to situations where a content is being stored or transferred from one location to the another. For example, in a peer-to-peer application, one of the above reactions may take place when the multimedia content is screened for watermarks at either the client or the server (or both). Finally, while the above system reactions have been introduced as separate categories, it is understood that there may be some overlap between the categories. For example, a particular reaction may be categorized both as prohibitive reaction and a status modification reaction. It is further understood the system response may comprise two or more of the above reactions.

Watermark Uncertainty Resolution

Prior to initiating one or more of the above described system reactions, it is necessary to resolve any uncertainties that may be associated with the detected watermarks. Such uncertainties may arise due to the detection of multiple watermark states, inadvertent capture of watermarks, presence of content with multiple credentials, or variations in content or system properties.

Detection of multiple watermark states: Multiple watermarks may be detected from a single content in any one of the following scenarios: due to a false detection, due to inadvertent

embedding of an already embedded content, or due to intentional embedding of an already embedded content. In a well-designed watermarking system, false detections, although possible, should be highly improbable. This probability can be even further reduced by triggering an enforcement action only when the same watermark is detected multiple times. Aside from false detections, some intentional and unintentional re-embedding attempts can be preempted by placing an in-line extractor within each embedder unit to detect the presence of pre-existing watermarks prior to any new embeddings. This approach was discussed earlier in the context of protection against overwriting attacks. Other than the simple cases listed above, there are other possibilities that may create bigger challenges for a watermarking system. For example, an intentional attack may be carried out by ‘mixing’ different versions of the same content (i.e., with different embedded watermarks) in order to confuse the extractor and possibly produce a reaction that is not consistent with the intended goals of the enforcement policy. It is also possible to unintentionally acquire multiple watermarks within a content. For example, a camcorder that is recording a home video (e.g., on the streets of New York, at a wedding ceremony, or at a birthday party) may capture different types of watermarked content from the surrounding environment. Another example may occur when multiple tracks from a playlist are selected and superimposed to legitimately create the final version of the content. If the selected tracks contain different types of watermarks, then the composite signal will also contain multiple types of watermarks, with possibly conflicting enforcement rules.

Given that the presence of multiple watermark states within a content signal is certainly possible, it is important for the extractor to actively continue its search for other possible watermarks. This could mean a thorough search of the entire watermarking space with high granularity even if the extractor is already tracking (e.g., is in synchronization with) one or more embedded watermarks.

One approach in dealing with the presence of multiple watermark states is to enforce the most restrictive set of rules associated with detected watermarks. For example, in the presence of a “no record” and a “copy freely” watermark state, the rules associated with the “no record” state would be enforced. This approach may be perfectly reasonable if, for example, a theatrical “No Home Use” watermark is detected in conjunction with any other watermark state. Alternatively, it may be desired to enforce the less restrictive action (or some other action all together), especially if the more restrictive watermark is not persistently

detected for a certain period of time. For example, a restrictive enforcement condition may only occur if the density and spacing of detected watermarks within an audio content reach the following thresholds: at least 10 watermarks are detected in each of 3 consecutive 7-minute segments of the audio. This condition provides a grace period of over 14 minutes and alleviates the concerns regarding an overly aggressive enforcement policy. In addition, the particular enforcement action and duration (as discussed under the heading "System Reactions to Copy Management Watermarks") may be selected in accordance with the detected watermark states, the density and distribution of such detections, the type of detection device, and the value of the content that is being protected. The logic of accumulating several detections (of the same state) before commencing an action was introduced earlier in the context of weight accumulation algorithm and time diversity buffers. Therefore, it is understood that the above discussion of the selection of enforcement action and duration is equally applicable to situations where only one type of watermark state is present. In such a scenario, there are two reasons why having a grace period may be beneficial. First, the reliability of extracted watermarks improve as more content is analyzed, and second, a harsh enforcement policy is avoided.

Inadvertent capture of watermarks: This scenario was described earlier as inadvertent capture of watermarked content through hand-held recording devices, such as camcorders or mobile phone (also referred to as the 'Birthday Party' scenario). In such cases, the uncertainty arises as to whether the captured content is the result of an illegal and intentional act, or an unintentional and momentary capture of a watermarked content. The above-described methods of examining the density and spacing of detected watermarks in multiple detection periods are certainly applicable for resolving this uncertainty. Another approach would be to include the 'quality' of detected watermarks as a factor in establishing whether further assessment of watermarks and/or a grace period is necessary. In other words, since acoustic/video capture of the content will inevitably result in some degradation of the embedded watermarks, the presence of high quality detections (e.g., watermarks that are detected with few erroneous symbols or missing components) is likely to preclude the possibility of such acoustic/video capture. It is further possible to design the watermarking system in such a way to identify the extent and type of signal modifications by examining a fragility profile of the extracted watermarks. For example, the embedded watermarks may contain certain components that are

destroyed completely, or degraded gracefully, as a result of acoustic/video capture. These and other techniques for evaluating possible signal modifications are described in the allowed U.S. Patent application No. 09/535,154 "Method and Apparatus for Detecting Processing Stages Applied to a Signal," assigned to the present assignee.

Presence of content with multiple credentials: In certain applications involving Digital Rights Management (DRM) schemes, a digital content may be subject to varying levels of watermark screening (and subsequent reactionary restrictions) based on the 'credentials' or trustworthiness of the content. For example, a content that is encrypted with 'algorithm A' may be freely transferred or recorded without watermark screening while an unencrypted content may be subject to watermark screening and possible restrictive reactions. In such systems, the uncertainty arises when content with different credentials are combined together to form a final version of the digital content. For example, a multimedia player may be configurable to accept and combine several input tracks, with different credentials, to produce a final content. A watermark extractor that is placed at the output of the 'mixer' may detect multiple watermark states and enact an inappropriate (either too permissive or too restrictive) or an inconsistent enforcement action. This situation can be remedied by screening each track (or groups of tracks with similar credentials) separately, and enacting the appropriate enforcement action at the input of the mixer. This way, each track, or group of tracks, is screened and reacted to in accordance with its credential. Thus, for example, a group of input tracks that are encrypted with 'algorithm A' may be allowed to pass through without watermark screening, while a second group of tracks that are encrypted with 'algorithm B' may be screened for watermarks, and subjected to any one of possible restrictive reactions in accordance with the enforcement rules of the extracted watermarks. In order to save computational resources, it may be desirable to trigger such screening and enforcement at the input of the mixer only upon the detection of multiple watermark states at the output of the mixer.

Variations in content or system properties: The watermark extractor may sometimes encounter intentional or unintentional changes in the content and system properties which may result in re-configuration of the extractor internal modules. Examples of variations in content and its properties include variations in signal sampling rates, compression techniques, source of content, etc. System level variations may comprise fast forwarding the content, pressing the record button while playing the content, etc. These variations may activate additional (or

different) components of the extractor. For example, if the input signal sampling rate is changed, a new resampling factor may have to be activated in the extractor. In this case watermark detection should be continued uninterrupted, with as little disturbance as possible (e.g. maintaining real time clock as close as possible). Exceptions to this rule may include actions that are seriously disruptive to user experience, such as disc tray open/close (where source of the content is physical medium on a disk), power off/on, etc.

Watermark Payload Expansion

The extent of payload capacity is often decided during the course of designing the watermarking system, and as disclosed earlier, it usually involves a trade off against other factors, such as the complexity of the detection process, the overall system security, and the robustness/reliability of detections. As it is often the case, at the completion of system design, payload capacity is finalized and cannot be changed any further. This approach works well for systems in which the extent of required payload is known. For example, if a watermark is used to carry the International Standard Recording Code (ISRC), it may require exactly 128 bits of payload capacity. In many applications, however, long term usage requirements of the watermarking system is uncertain, and new watermark states may be necessary for future needs. The typical approach under these circumstances is to include several "reserved" or unused fields to accommodate future needs of the watermarking system. It is also often the case that these future applications never materialize. Thus this accommodation for probable future needs comes at the expense of system security, robustness and extraction complexity of the present system.

It is therefore an object of the present invention to design a watermarking system to accommodate the future needs of a watermarking system without sacrificing robustness, computational complexity or security of the present system. This can be accomplished by employing a multi-tier watermarking technique. The first (or base) tier is designed to produce maximum robustness and security within the limitations of system resources. The base tier also carries a fixed payload in accordance with the needs of the current watermarking applications. For example, the base tier may have the capacity to carry 4 different watermark states (i.e., a two-bit payload). Similarly, the extractor is capable of detecting any one of four embedded watermark values/states.

As described previously in the example embodiments of the present invention, during the embedding process, various embedding opportunities are identified within the host content in accordance with multiple embedding algorithms, embedding frequency ranges, embedding time slots, PN sequences, or other parameters that comprise the stego key (for example, see Fig. 3). In designing the base watermark tier, all such embedding opportunities may carry the same watermark payload information. It is, however, possible to increase the payload capacity, at a future time, by allocating a first set of embedding opportunities to a first set of payload values, and a second set of embedding opportunities to a second set of payload values, and a third set of payload values to a third set of embedding opportunities, and so on. This way, a single tier watermarking system is converted into a multi-tier watermarking system, where each tier comprises a particular set of embedding opportunities, and each tier comprises a set of payload values that is, at most, as large as the original single tier (i.e., the base tier). While each watermarking tier by itself may not exceed the payload capacity of the original system, when two or more tiers are combined together, an increase in payload capacity is realized. This concept is further illustrated by the example disclosed herein.

In the most simple example, this technique translates into a form of time-division multiplexing, where payload values embedded in two neighboring time intervals are viewed together to form the increased payload set. In other words, a first payload value is embedded in the first, third, fifth, ... Nth, time slots while a second payload value is embedded into the second, fourth, sixth, ... Nth+1, time slots. A pair of time slots, taken together, would then represent an expanded payload code space. This concept is illustrated in Fig. 23, where a 4-state (i.e., 2-bit) payload space is expanded into a 10-state payload space by considering pairs of original codes together. The foregoing example may also be modified to provide an even larger payload space by viewing three, four, five, ... codes together. Note that in the example embodiment of Fig. 23, the order of appearance of the 2-bit codes is not relevant (e.g., the codewords (00,10) and (10,00) are not distinguished from one another); These codewords are designated as "repeat codewords" in Fig. 23. This consideration is important for facilitating the detection of watermarks in most watermarking systems, where it is not practical to keep track of the order of embedded codewords. However, in systems where the order of embeddings are known (and can be conveyed to the extractor), these pairs may be used to further expand the payload space. In some applications, it may be advantageous to preclude certain original codes

from participating in the payload expansion scheme. For example, in the example embodiment of Fig. 23, one may decide to forbid combinations of the code '11' with any code other than , itself, thus prohibiting the embedding of the states (00,11), (11,00), (01,11) (11,01), (10,11) and (11,10) to occur. This exclusion removes any ambiguities associated with the detection of the code '11'; i.e., the detection of '11' unequivocally signals the presence of one, and only one, possible watermark state. It is further possible to generalize this concept by expanding the code space using multiple embedding algorithms, frequency bands, or other components of the stego key. In other words, each constituent component of the embedding stego key may be used as an additional dimension to expand the payload space.

There is a price associated with an increase of the payload capacity. First, it is necessary to update the extractors to enable the interpretation of the new code space. This upgrade may not be feasible for a category of devices with no upgrade capability. On the other hand, this modification does not involve a major design change or increase in computational resources. It only suffices to convey the new interpretation of detected payloads to the extractor (e.g., in the form of a lookup table). To this end, the original extractors may be equipped with pre-existing 'hooks' to enable seamless integration of the new payload table. Another penalty associated with expanding the payload is either increased detection time or decreased robustness of detections. More specifically, since the set of all embedding opportunities are now divided – not necessarily equally – between two or more set of codes, it may take a longer period of time to reliably detect a given watermark state; or alternatively, the watermarks from a fixed time interval may be detected with a lower reliability simply because there are fewer of them in that interval. The latter can also be viewed from a different perspective: it may take less of an effort for an attacker to interfere with proper interpretation of extracted watermarks since it suffices to remove half (or a subset) of all embedded watermarks. For example, in the time-division multiplexing example of Fig. 23, this may be accomplished by obliterating the embedded watermarks from only the first, third, fifth, ...Nth time intervals. Although this removal/jamming technique may not be feasible in more complex systems, it nevertheless reduces the robustness of watermarks.

This "vulnerability" may also be advantageously used to incorporate new features into the system. For example, it is possible to intentionally design a multi-tier watermarking system with tiers of unequal robustness. In these systems, while the detection of all tiers may indicate

a certain watermark state, the detection of a single tier (or the absence of a single tier) would be interpreted as a different watermark state. This particular type of multi-tier system has applications in both tamper detection and copy protection areas. For example, the absence of a less robust set of watermarks may signal that the original content has been tampered with. In a copy protection environment, this may trigger a set of enforcement actions that are different, and perhaps more restrictive, than the enforcement actions associated with the original multi-tier state. The degree of robustness/fragility of each watermark tier may be adjusted by allocating a smaller or larger proportion of embedding opportunities to that watermark tier. In addition to the allocation of the *number* of embedding opportunities, fragility characteristics may be more appropriately designed by allocating certain *types* of embedding opportunities to a particular watermark tier. For example, in an audio watermarking application, one watermark tier may be designed to occupy the frequency range 8-20 KHz that is susceptible to highpass filtering; the absence of this watermark tier in a received content would then signal the likelihood of a highpass filtering operation. Similarly, certain embedding algorithms may provide better protection against one type of attack on the host content while provide very little protection against others. In such a case, the failure of that particular detection algorithm to extract sufficient number of watermarks can signal the presence of a certain type of attack or tampering. In general, watermarking opportunities can be divided between two or more watermark tiers where each tier is tailored to have an *overall* level of robustness/fragility as well as *specific* degrees of robustness/fragility.

It should be noted that the expanded payload systems disclosed above maintain perfect backward compatibility with the content embedded with the original payload configuration. Thus, unless otherwise desired, such a system provides the original levels of robustness and security for content that is only embedded with a base tier. Furthermore, the detection time/robustness penalty may be fully or partially compensated by increasing the embedding strength. This option, however, may not be desirable in many applications where transparency of watermark embedding is of utmost importance.

Multi-Channel Selection and Mixing Attacks

In systems with limited computational resources, it may not be feasible to examine all input multimedia channels for the presence of watermarks. This may be especially true for an

audio track, where 16 or more audio channels may be present at any given time. In accordance with the present invention, one method of achieving computational feasibility in screening all input audio channels is to mix a subset of all available input channels into a single monophonic audio signal before screening for the presence of embedded watermarks. Equation 20 describes channel mixing in mathematical terms, where 'n' channel are mixed together to form a single channel, C.

$$C = a_1.c_1 + a_2.c_2 + a_3.c_3 + \dots + a_n.c_n. \quad \text{Equation (20)}$$

The coefficients $a_1, a_2, a_3, \dots, a_n$ ($0 \leq a_n \leq 1$) are weights that multiply the individual channels; these coefficients may be constant valued, or may vary as a function of time, frequency, amplitude, energy or other variables that are dependant or independent of the accompanying c_n channels. In a 5.1 audio format, for example, these coefficients may be selected differently for the Left and Right channels as opposed to the remaining audio channels. The coefficients may also be selected based on inherent characteristics of the associated channels. For example, an audio channel may be examined for the presence of special characteristics, such as pre-defined amplitude variations, frequency distribution, energy profile and others, before a proper coefficient value is selected. This examination, analogous to a psycho-acoustic or psycho-visual analysis, may reveal whether or not a particular channel is inherently suitable for carrying watermarks before selecting the corresponding weight coefficient (e.g., if embedded watermarks are known to occupy the lower range of audio frequencies but the selected channel only contains a small range of such frequencies, a small weight coefficient may be selected). In addition, such an analysis may reveal whether or not a particular channel contains dummy information or test signals, and therefore should be assigned a lower weight (or excluded from consideration all together). For example, a channel that is comprised of purely white noise or pure silence may qualify for such consideration.

To further improve the extractor robustness in the presence of intentional attacks, (a) a subset of all available input channels may be selected, (b) such selection may vary in time, and (c) such selection may be done probabilistically. These collective steps help introduce additional uncertainty and variability in the extraction process and thus prevent an attacker from formulating a consistent and effective attack. For example, in a 16-channel audio file

(n=16), there are $\sum_{k=1}^{16} \binom{k}{n}$ or 65,535 different combination of channels that can be selected.

Furthermore, each selection may last for a t -second duration ($T1 \leq t \leq T2$), and each channel combination may be selected with a probability p ($0 \leq p \leq 1$). The particular value of the selection duration lower bound, $T1$, is selected based on the minimum duration required for reliable extraction of embedded watermarks, but may otherwise be arbitrary. The selection of the upper bound of selection duration, $T2$, involves reaching a balance between the desired levels of robustness and the available computational resources. Furthermore, t may be a fixed value that is selected by the system designer for all extractors, it may be a fixed value for each individual extractor, or may randomly vary between $T1$ and $T2$, for each channel selection interval and for all extractors. The probability of channel selection, p , may be selected to follow a uniform probability distribution, or may be modified based on other factors, such as the number of available channels, the history of a particular channel combination (i.e., whether or not a particular channel combination has previously produced meaningful watermark detections), or other factors which may favor the selection of one set of channel combinations over the others. These considerations are also applicable when channel weight coefficients, a_n , are selected.

Although the invention has been described in the context of various preferred embodiments, it should be appreciated that many different adaptations of the present invention may be made without departing from the scope of the invention. For example, the techniques describes in the present invention may be readily adapted to analog, digital, optical or acoustical domains. This includes, but not limited to, the utilization of optical and acoustical techniques for manipulating the signals of present invention. Additionally, the "signals" described in the context of present invention refer to any entity that can be manipulated to effect the various embodiments of the present invention, ranging from electrical, electromagnetic or acoustic signals to the signals produced by mechanical shaping of a surface. The latter, for example, may be the plastic layer that covers optical storage media or the laminate that covers a driver's license. Furthermore, the signals of the present invention may be transmitted, displayed or broadcast or may be stored on a storage medium, such as an optical or magnetic disk, an electronic medium, a magnetic tape, an optical tape or a film.

What is claimed is:

1. A method for adapting the operation of a watermark extractor in response to watermarks detected in a digital host signal, comprising:
 - (a) receiving said digital host content;
 - (b) detecting embedded watermarks in said digital host content;
 - (c) analyzing the detected watermarks to determine whether any of said detected watermarks comprise strong watermarks; and
 - (d) if none of the detected watermarks comprise a strong watermark, modifying an operation of the extractor if at least one weak watermark is detected.
2. The method of claim 1, wherein said modifying comprises selecting a different extraction stego key.
3. The method of claim 1, wherein said modifying comprises extending the extractor operation to search at least one additional:
 - pixel area of said digital host content;
 - time segment of said digital host content;
 - spatial frequency range of said digital host content;
 - temporal frequency range of said digital host content;
 - color component of said digital host content;
 - audio channel of said digital host content; and
 - component of a compressed version of said digital host content.
4. The method of claim 1, wherein said modifying comprises increasing a granularity of watermark detection.
5. The method of claim 1, wherein said modifying comprises at least one of
employing additional error correction code techniques; and
conducting forensic analysis.

6. A method for adapting the operation of a system in response to watermarks detected in a digital host content, said method comprising:

receiving said digital host content;

detecting embedded watermarks in said digital host content;

analyzing the detected watermarks;

adapting the operation of the system in accordance with said analyzing step by at least one of:

permitting normal operation of said system;

conditionally permitting the operation of said system;

prohibiting the normal operation of said system;

modifying a status of said digital host content; and

degrading a quality of said digital host content.

7. The method of claim 6, wherein said conditionally permitting comprises permitting the operation of said system provided a response from a user, a database, or a third party is received in response to a request from said system.

8. The method of claim 6, wherein said prohibiting comprises at least one of

muting an audio portion of said digital host content;

blanking a video portion of said digital host content; and

stoppage of playback, recording or transfer of said digital host content.

9. The method of claim 6, wherein said modifying the status comprises changing the watermarks of said digital host content.

10. The method of claim 6, wherein said modifying the status comprises changing credentials associated with said digital host content.

11. The method of claim 6, wherein said modifying the status comprises reversibly obscuring said digital host content.

12. The method of claim 6, wherein said degrading comprises at least one of down-sampling, lossy compression, dynamic range reduction, partial scrambling, spectral shaping, addition of wow and flutter, addition of noise, phase distortion, intermittent blanking, and muting of said digital host content.

13. The method of claim 6, wherein said degrading is reversible.

14. The method of claim 6, wherein said degrading is not reversible.

15. A method for adapting the operation of a system in response to the detection of multiple watermark states from a digital host content, comprising:

receiving said digital host content;

detecting embedded watermarks in said digital host content;

analyzing the detected watermarks to determine a watermark state of each of said detected watermarks, each watermark state having a distinct enforcement rule associated therewith; and

selecting one of said enforcement rules associated with said detected watermarks for use in operation of the system.

16. A method in accordance with claim 15, wherein:

the most restrictive enforcement rule associated with said detected watermarks is selected.

17. A method in accordance with claim 15, wherein

the least restrictive enforcement rule associated with said detected watermarks is selected.

18. A method in accordance with claim 15, wherein:

the watermarks are detected from at least two monitoring intervals; and

an enforcement action is commenced if at least a first watermark state is detected from two or more monitoring intervals, said enforcement action corresponding to said enforcement rule associated with the first watermark state.

19. The method of claim 18, wherein said enforcement action comprises at least one of:

- permitting normal operation of said system,
- conditionally permitting the operation of said system;
- prohibiting the normal operation of said system;
- modifying a status of said digital host content; and
- degrading a quality of said digital host content.

20. The method of claim 18, wherein said first watermark state is detected in accordance with a predefined value, type, density or spacing.

21. The method of claim 18, wherein said enforcement action is extended in response to the detection of each additional watermark state.

22. The method of claim 18, wherein a duration of said enforcement action is selected randomly.

23. The method of claim 18, wherein a duration of said monitoring intervals is selected randomly.

24. A method for reducing the computational complexity of watermark extraction from a multi-channel digital host content, comprising:

- receiving a multi-channel digital host content;
- selecting a subset of received channels of said multi-channel digital host content;
- combining the selected subset of received channels to form a composite signal; and
- extracting embedded watermarks from said composite signal.

25. The method of claim 24, wherein said selecting is done in accordance with a probability value.

26. The method of claim 25, wherein said probability value has a uniform distribution.

27. The method of claim 25, wherein said probability value has a non-uniform distribution.

28. The method of claim 25, wherein said probability value is calculated in accordance with a number of channels in said multi-channel digital host content.

29. The method of claim 23, wherein:

- a new subset of received channels is selected after a time interval;
- the channels from the selected new subset are combined to form a new composite signal; and
- the watermarks are extracted from the new composite signal.

30. The method of claim 29, wherein a duration of said time interval is random.

31. The method of claim 24, wherein said combining comprises:

- obtaining a coefficient associated with each selected channel in said subset; and
- adding together the selected channels in said subset in accordance with the associated coefficients.

32. The method of claim 31, wherein said coefficients are equally valued.

33. The method of claim 31, wherein said coefficients are selected in accordance with characteristics of the selected channels in said subset.

34. The method of claim 31, wherein said coefficients are selected in accordance with a probability value.

1/ 21

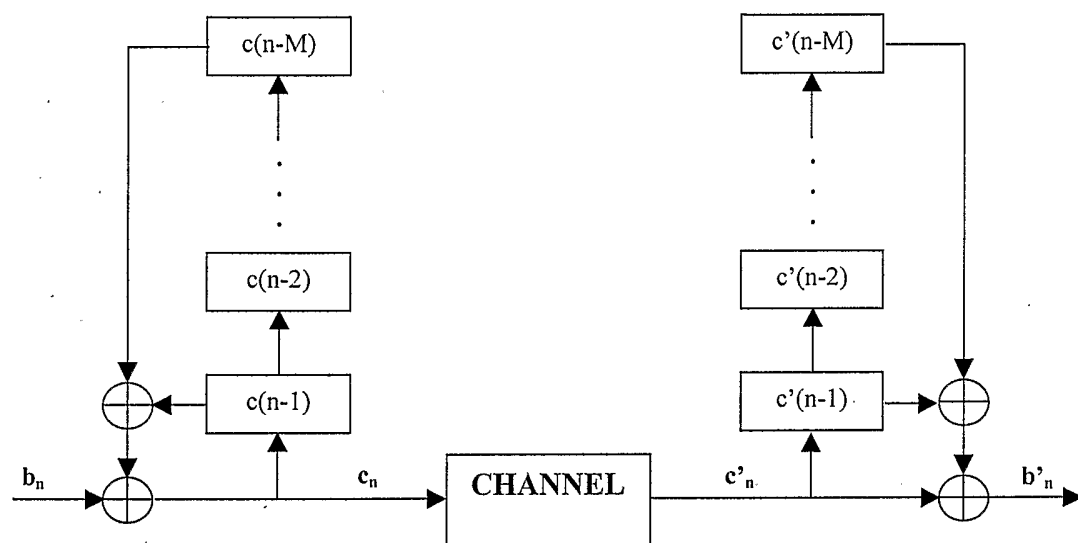


FIG. 1

2/ 21



FIG. 2

Dimension Index (i)	Description Of Component	Component Key Length (L_e^i)	
1	Time segment	16	
2	Type of Algorithm	4	
3	Frequency shift	27	
4	PN sequence	10	
5	Frequency band	25	Key Length (L_e) = 82 bits

FIG. 3

3/ 21

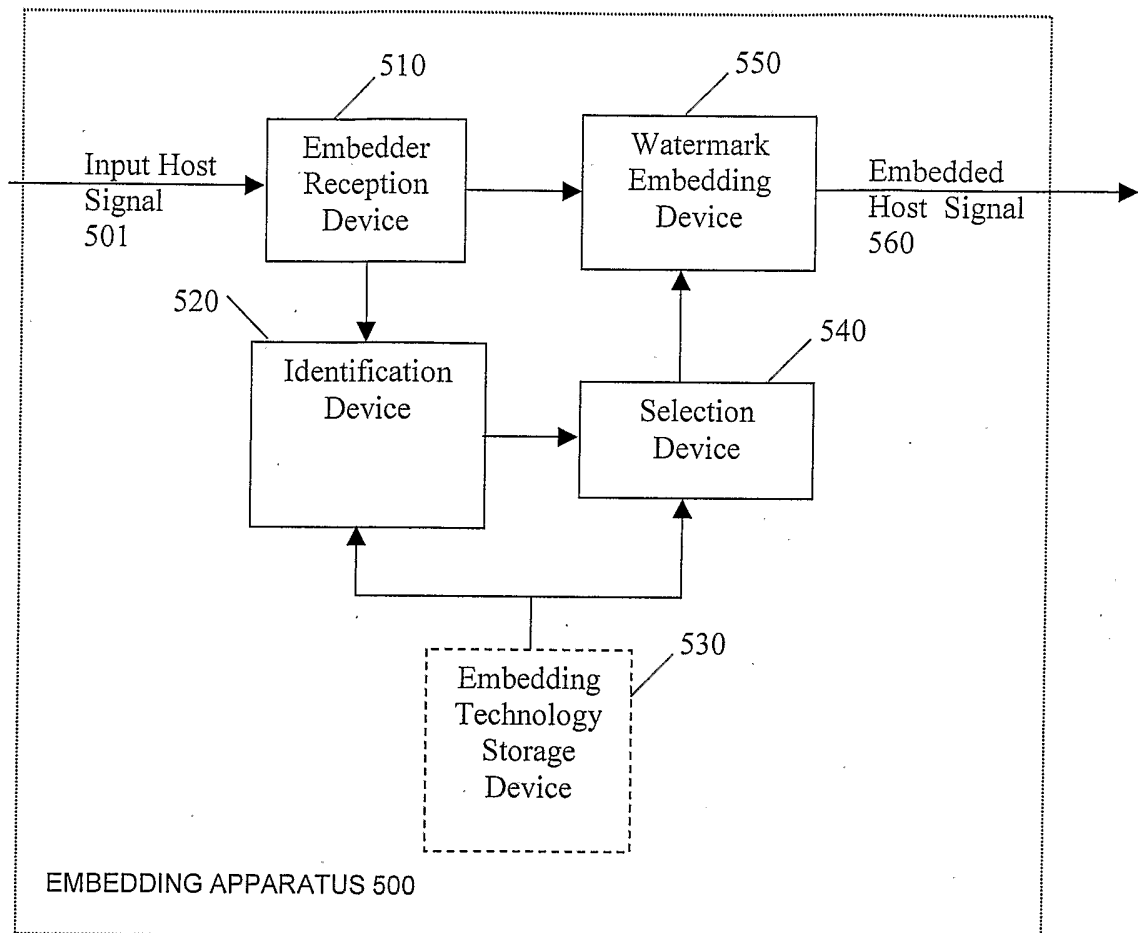


FIG. 4

4/ 21

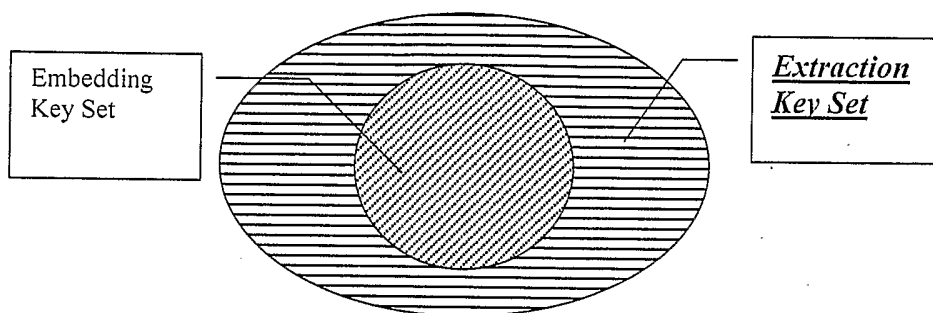


FIG. 5a

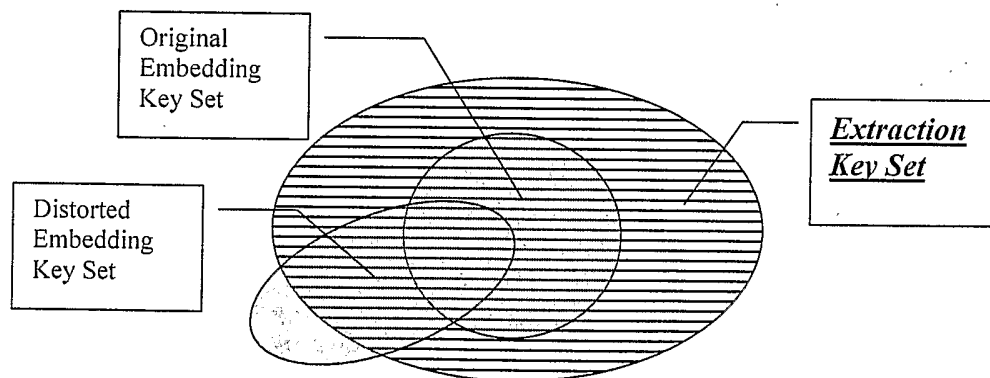


FIG. 5b

5/ 21

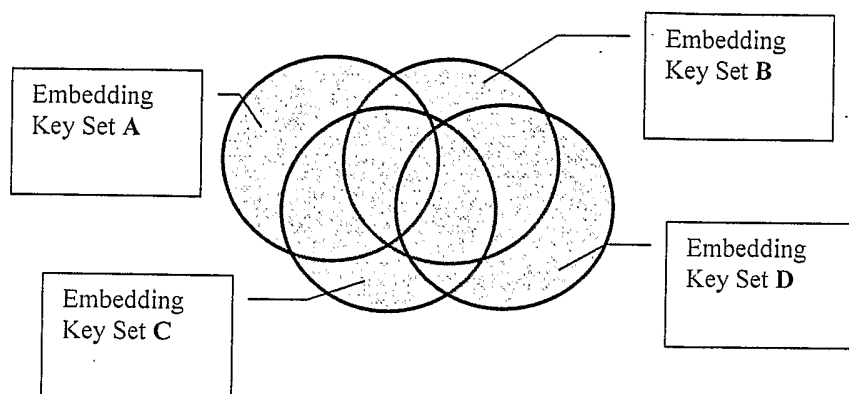


FIG. 6a

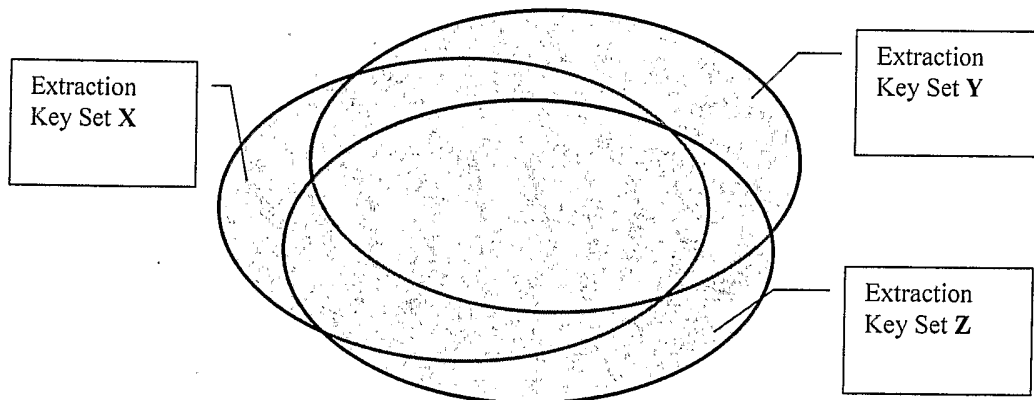
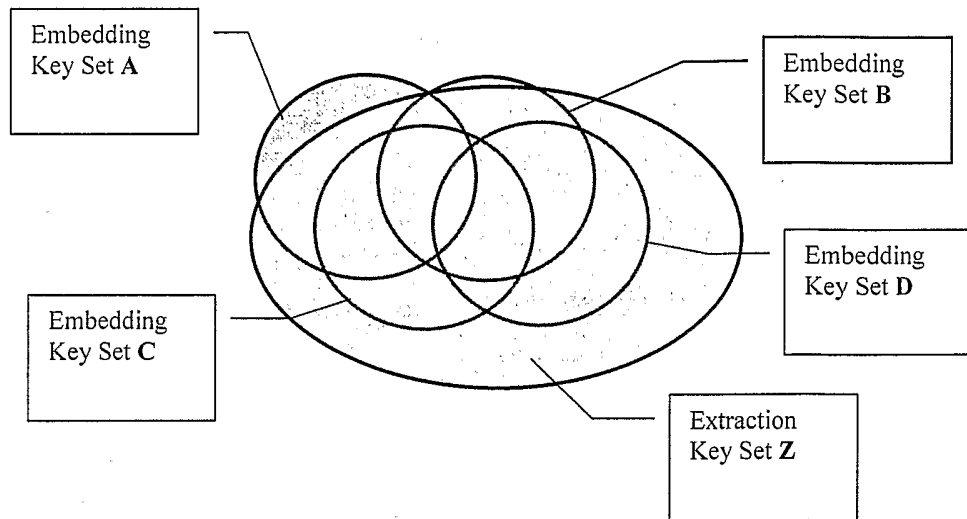


FIG. 6b

6/ 21

**FIG. 6c**

7/ 21

	Robustness	Security	Transparency	Computational Expense	False Detections
Increasing embedding key length	Improves	Improves	Degrades	Increases	Increases
Increasing Extraction Key length	Improves	Improves	N/A	Increases	Increases
Increasing Watermark Strength	Improves	Somewhat Degrades	Degrades	N/A	N/A
Increasing Overlap Between Detection Key Sets	N/A	Degrades	N/A	N/A	Decreases

FIG. 7

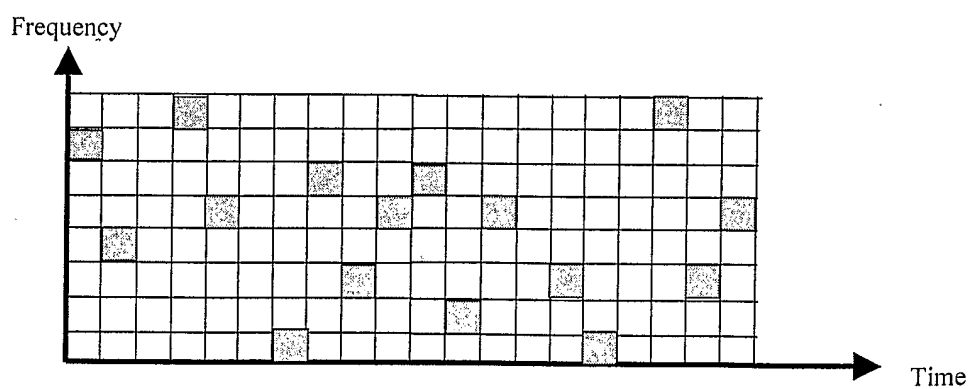


FIG. 9

8/ 21

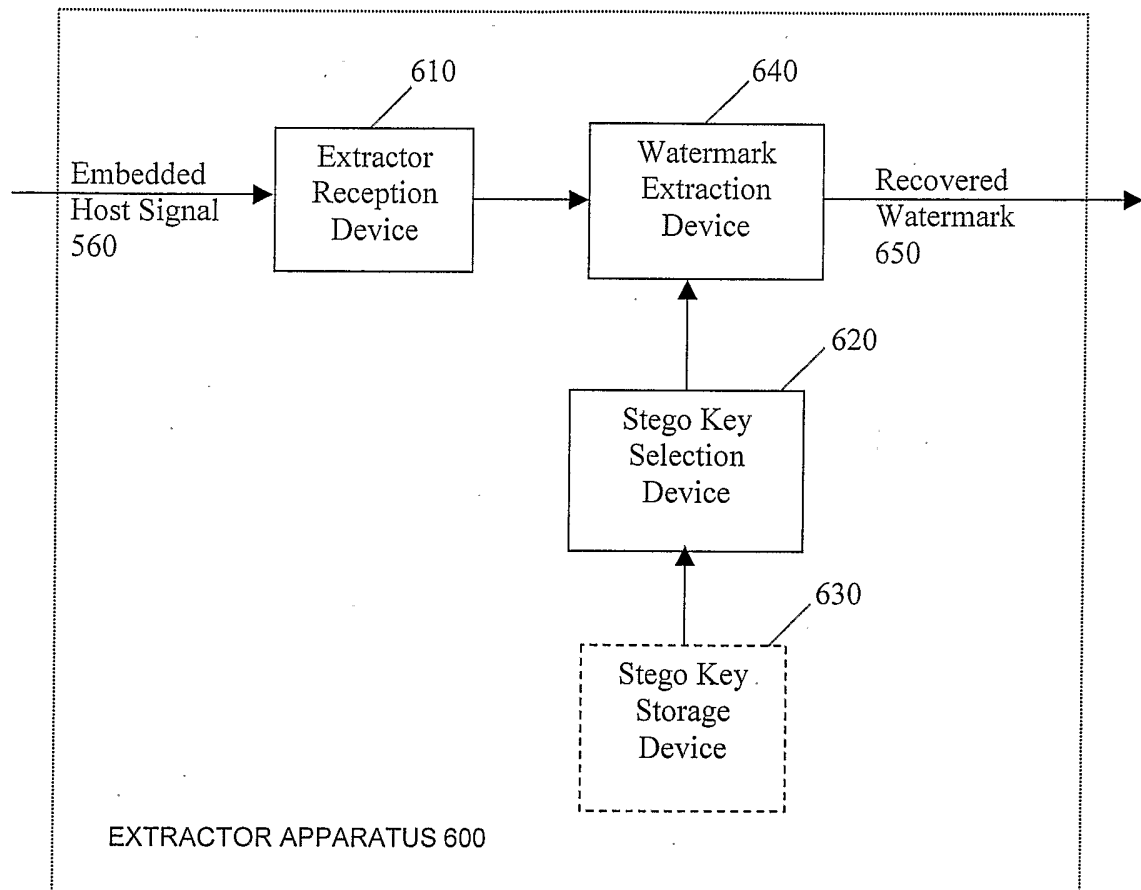


FIG. 8

9/ 21

	Content #1	Content #2	Content #3	Content #4	Content #5	Content #6
1	<u>11271</u>	<u>11271</u>	5663	4163	<u>11271</u>	<u>11271</u>
2	13809	14063	<u>11271</u>	<u>11271</u>	21023	13809
3	14063	35207	13809	19590	35720	20871
4	31568	35720	<u>37743</u>	<u>37743</u>	<u>37743</u>	24002
5	35720	<u>37743</u>	59768	50971	51658	30980
6	<u>37743</u>	52668	<u>60032</u>	<u>60032</u>	<u>60032</u>	<u>37743</u>
7	46468	56720	62314	63108	<u>67214</u>	42689
8	48050	<u>60032</u>	<u>67214</u>	<u>67214</u>	<u>69166</u>	<u>60032</u>
9	<u>60032</u>	<u>67214</u>	<u>69166</u>	<u>69166</u>	71333	<u>67214</u>
10	<u>67214</u>	67978	<u>72694</u>	<u>72694</u>	<u>72694</u>	<u>69166</u>
11	<u>69166</u>	<u>69166</u>	<u>78029</u>	<u>78029</u>	<u>78029</u>	<u>72694</u>
12	69209	<u>72694</u>	79277	79988	84627	76388
13	<u>72694</u>	<u>78029</u>	85817	<u>99959</u>	90467	<u>78029</u>
14	74218	78847	86245	105448	92737	81732
15	<u>78029</u>	83796	90852	118214	96293	95673
16	82864	84627	92494	137812	96348	<u>99959</u>
17	84627	94271	94568	<u>143379</u>	<u>99959</u>	130304
18	94568	<u>99959</u>	96348	<u>149367</u>	106612	<u>143379</u>
19	<u>99959</u>	129792	98220	<u>152362</u>	113817	143742
20	105448	131136	<u>99959</u>	<u>159072</u>	126920	148938
21	132881	<u>143379</u>	117085	<u>160711</u>	<u>143379</u>	<u>149367</u>
22	<u>143379</u>	<u>149367</u>	119970	167976	<u>149367</u>	<u>152362</u>
23	148852	<u>152362</u>	137374	<u>170526</u>	<u>152362</u>	<u>159072</u>
24	<u>149367</u>	<u>159072</u>	<u>143379</u>	172306	<u>159072</u>	<u>160711</u>
25	152352	160010	<u>149367</u>	194178	<u>160711</u>	161361
26	<u>152362</u>	<u>160711</u>	<u>152362</u>	<u>195009</u>	<u>170526</u>	168239
27	<u>159072</u>	<u>170526</u>	<u>159072</u>	201814	182391	<u>170526</u>
28	<u>160711</u>	173505	<u>160711</u>	<u>205656</u>	<u>195009</u>	172673
29	<u>170526</u>	186921	<u>170526</u>	213166	198459	173952
30	187039	188239	174875	231124	201094	191458
31	194178	<u>195009</u>	185818	<u>231833</u>	<u>205656</u>	<u>195009</u>
32	<u>195009</u>	200352	<u>195009</u>	238600	211093	198277
33	198955	<u>205656</u>	200678	<u>245871</u>	219890	<u>205656</u>
34	<u>205656</u>	209130	<u>205656</u>	<u>253456</u>	221726	210943
35	209439	214700	210013	253902	229452	211002
36	228647	217669	211093	<u>255341</u>	<u>231833</u>	228647
37	<u>231833</u>	227069	228647	268375	<u>245871</u>	<u>231833</u>
38	232076	228091	<u>231833</u>	280430	249497	<u>245871</u>
39	238600	228336	233418		<u>253456</u>	<u>253456</u>
40	<u>245871</u>	<u>231833</u>	<u>245871</u>		<u>255341</u>	<u>255341</u>
41	<u>253456</u>	<u>245871</u>	249841		278485	258140
42	<u>255341</u>	250153	250898		280788	261035
43	266893	<u>253456</u>	<u>253456</u>		282035	
44	268986	<u>255341</u>	<u>255341</u>		285816	
45		278485	276980		286023	

FIG. 10

10/ 21

	Extractor #1	Extractor #2	Extractor #3	Extractor #4	Extractor #5
1	60	36239	4163	21023	11271
2	3410	37743	37743	37743	11811
3	11271	38314	50971	51658	20871
4	21710	58831	58831	69166	24002
5	22723	59768	63108	71333	30980
6	25656	60032	72694	90467	37743
7	26727	62314	79988	92737	42689
8	27139	67214	99959	96293	57063
9	31568	73080	118214	99959	64802
10	36239	78561	137812	106612	69166
11	37743	79277	159072	113817	72180
12	46468	82940	160711	126920	73080
13	48050	85817	167976	149367	76388
14	54923	86245	171088	159072	100812
15	67214	98220	172306	160711	105101
16	69166	98644	178739	170526	108676
17	69209	101125	180459	195009	115957
18	78029	108676	182160	198459	120585
19	80924	117085	201814	201094	130304
20	82864	119970	211961	229452	132783
21	98644	132783	213166	249497	143379
22	101125	159122	215119	255341	143742
23	111873	160711	231124	280788	152362
24	125752	163196	231781	282035	160711
25	132881	171088	231833	285816	161361
26	140173	174177	253456	286023	163196
27	140201	174875	253902		166522
28	144565	176215	255341		168239
29	148852	183264	268375		170526
30	149367	185818	280430		173952
31	152352	195009			190886
32	159072	198878			191458
33	159122	200678			194880
34	160711	210013			198277
35	174177	211961			199954
36	179065	215119			203707
37	193492	219927			205656
38	198955	220404			210943
39	209439	231781			219536
40	209476	233418			225039
41	216531	245871			253456
42	231833	249841			261035
43	232076	250898			269327
44	232609	276980			270329
45	254420				
46	266893				

FIG. 11

11/ 21

	Extractor #1	Extractor #2	Extractor #3	Extractor #4	Extractor #5
1	60	36239	4163	21023	11271
2	3410	37743	37743	37743	11811
3	11271	38314	50971	51658	20871
4	21710	58831	58831	69166	24002
5	22723	59768	63108	71333	30980
6	25656	60032	72694	90467	37743
7	26727	62314	79988	92737	42689
8	27139	67214	99959	96293	57063
9	31568	73080	118214	99959	64802
10	36239	78561	137812	106612	69166
11	37743	79277	159072	113817	72180
12	46468	82940	160711	126920	73080
13	48050	85817	167976	149367	76388
14	54923	86245	171088	159072	100812
15	67214	98220	172306	160711	105101
16	69166	98644	178739	170526	108676
17	69209	101125	180459	195009	115957
18	78029	108676	182160	198459	120585
19	80924	117085	201814	201094	130304
20	82864	119970	211961	229452	132783
21	98644	132783	213166	249497	143379
22	101125	159122	215119	255341	143742
23	111873	160711	231124	280788	152362
24	125752	163196	231781	282035	160711
25	132881	171088	231833	285816	161361
26	140173	174177	253456	286023	163196
27	140201	174875	253902		166522
28	144565	176215	255341		168239
29	148852	183264	268375		170526
30	149367	185818	280430		173952
31	152352	195009			190886
32	159072	198878			191458
33	159122	200678			194880
34	160711	210013			198277
35	174177	211961			199954
36	179065	215119			203707
37	193492	219927			205656
38	198955	220404			210943
39	209439	231781			219536
40	209476	233418			225039
41	216531	245871			253456
42	231833	249841			261035
43	232076	250898			269327
44	232609	276980			270329
45	254420				
46	266893				

FIG. 12

12/ 21

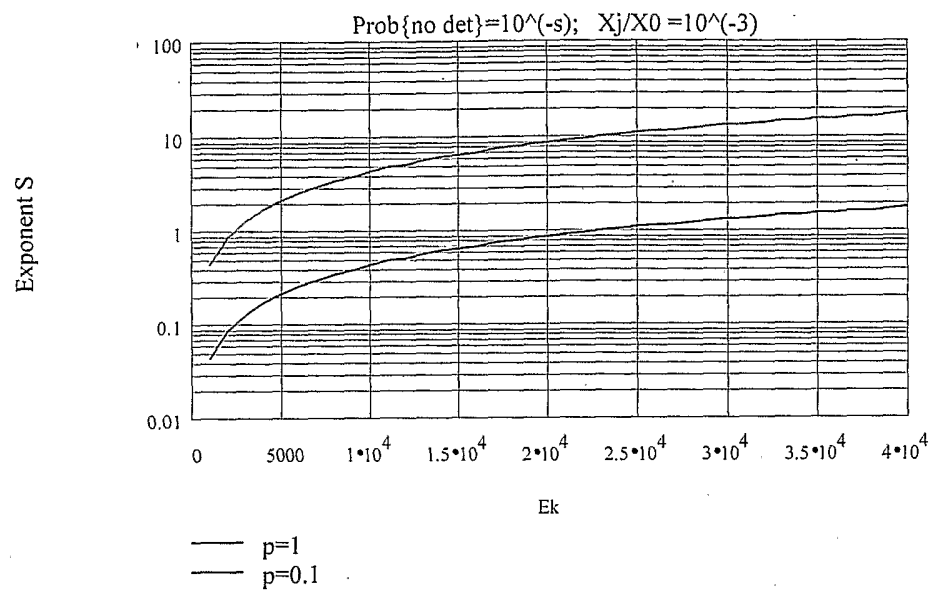


FIG. 13

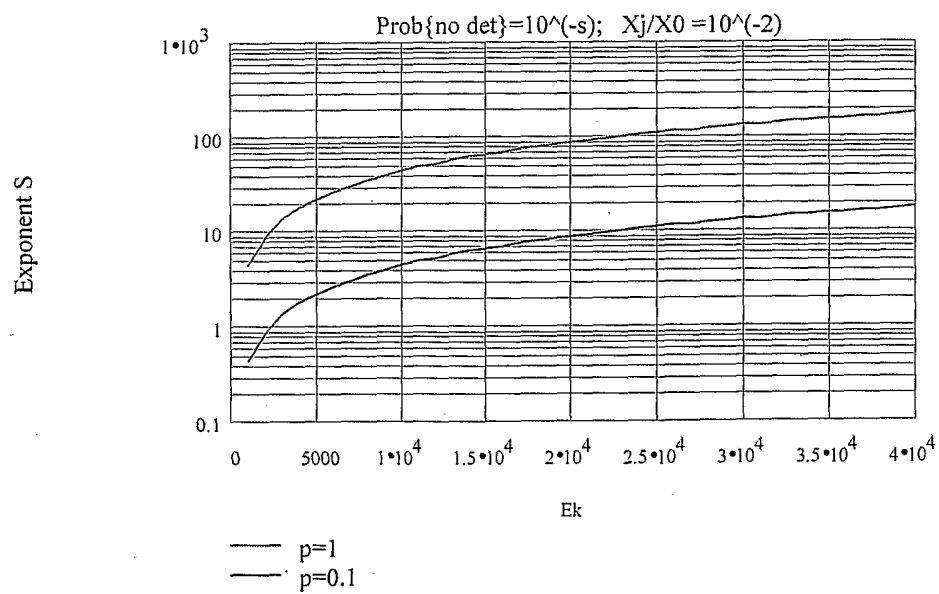


FIG. 14

13/ 21

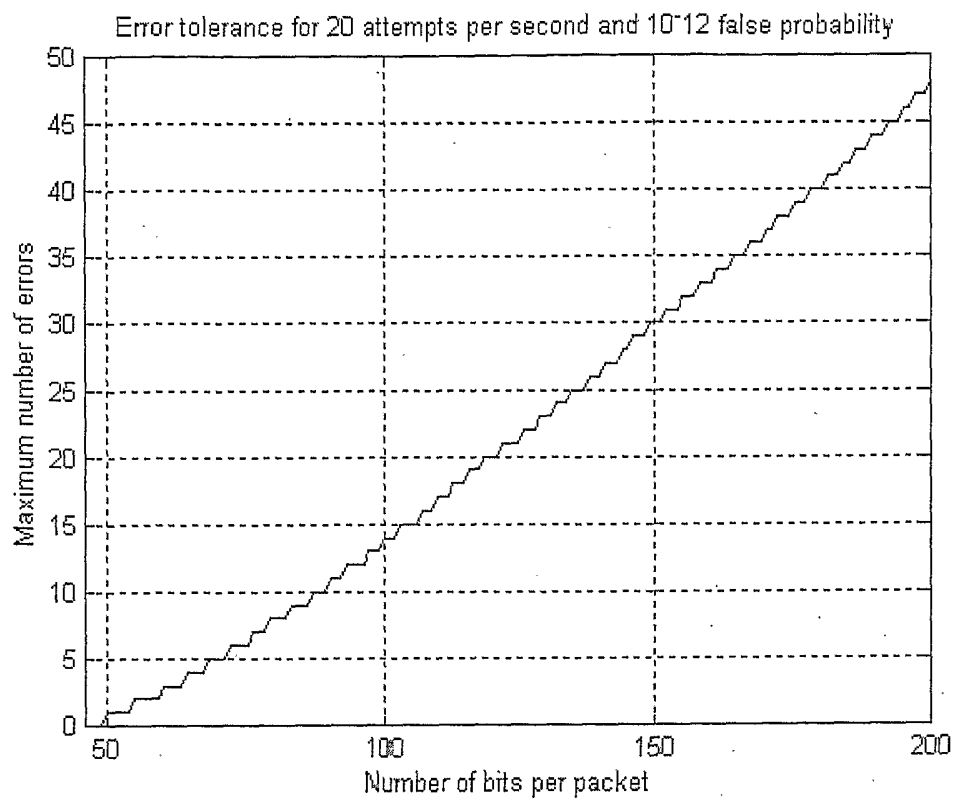


FIG. 15

14/ 21

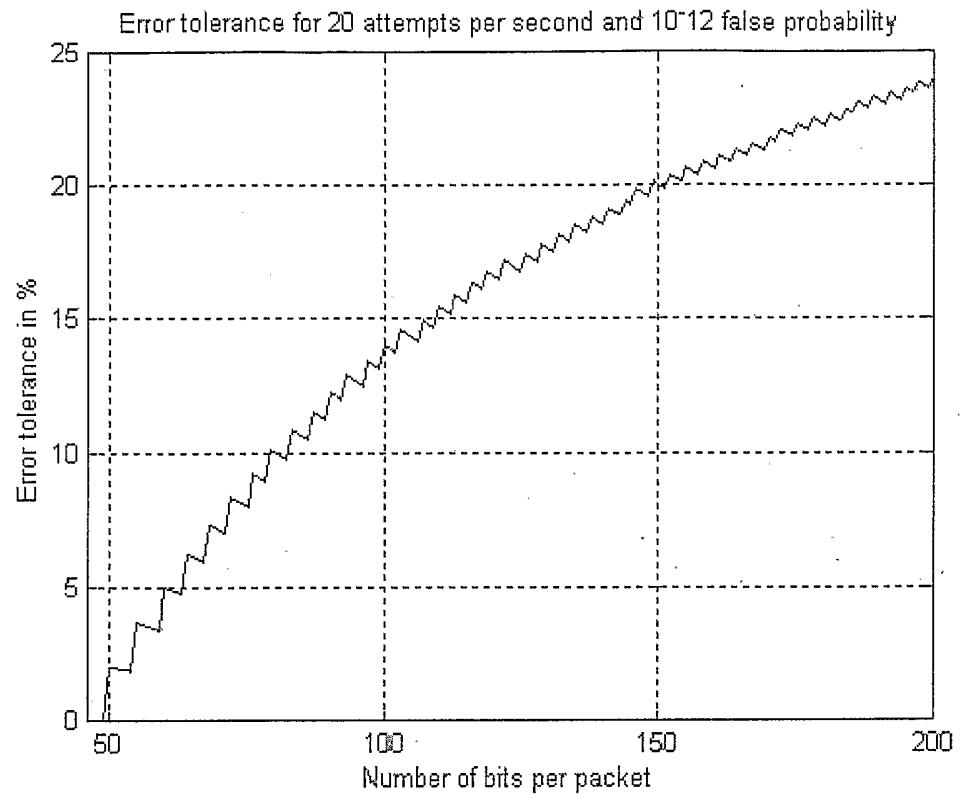


FIG. 16

15/ 21

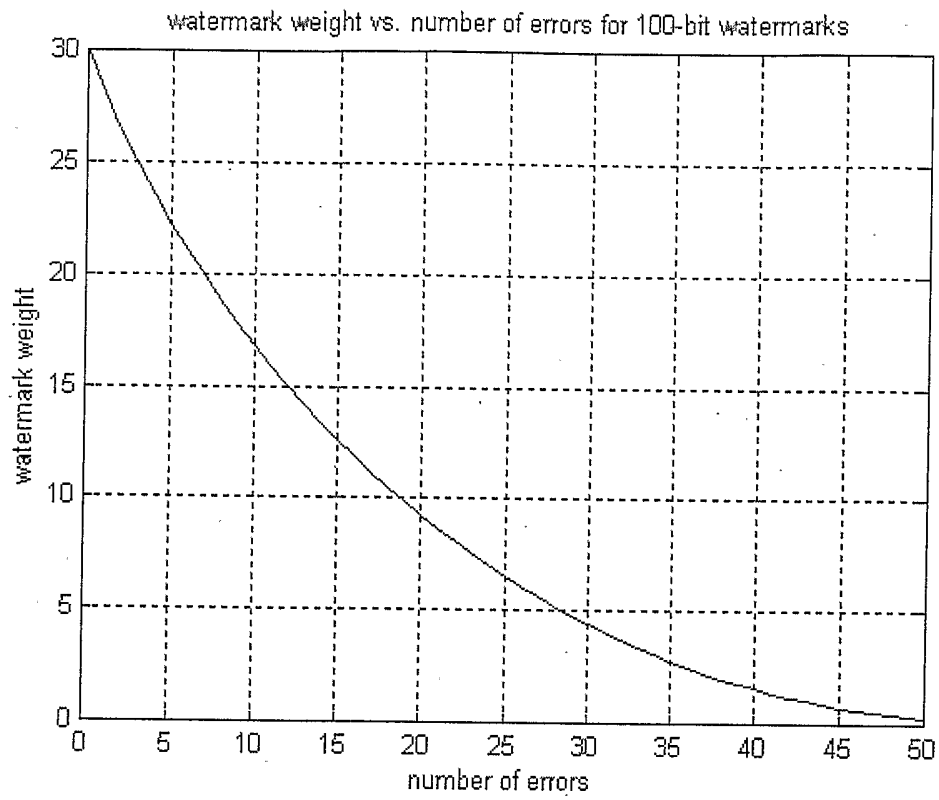


FIG. 17

16/ 21

Correlation Coefficient Range	Detected Bit Value	Probability Value
0.7 to 1.0	1	0.9
0.5 to 0.7	1	0.8
0.3 to 0.5	1	0.7
0.1 to 0.3	1	0.6
0 to 0.1	1	0.55
-0.1 to 0	0	0.55
-0.3 to -0.1	0	0.6
-0.5 to -0.3	0	0.7
-0.7 to -0.5	0	0.75
-1.0 to -0.7	0	0.9

FIG. 18

17/ 21

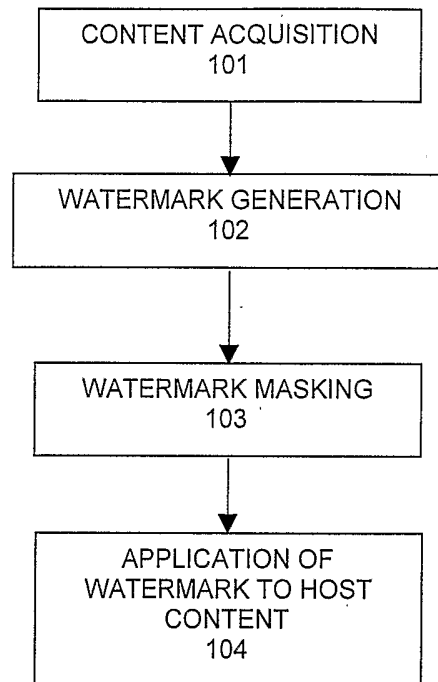


FIG. 19

18/ 21

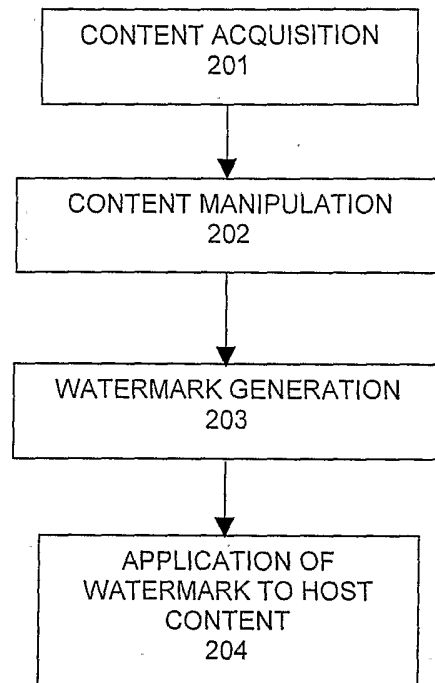


FIG. 20

19/ 21

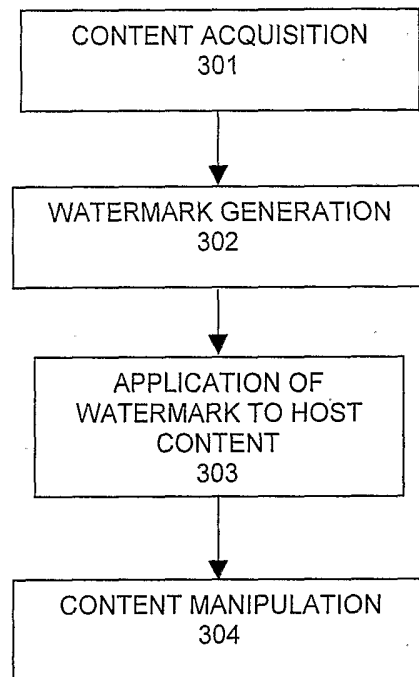


FIG. 21

20/ 21

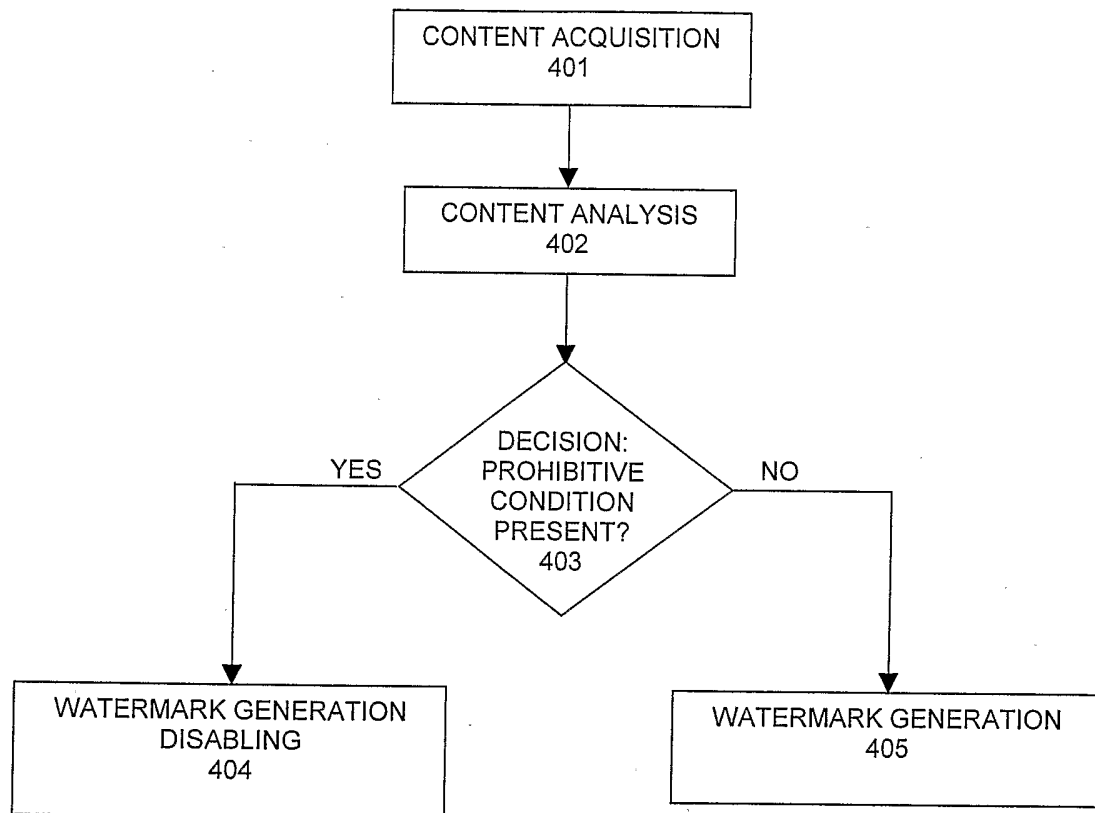


FIG. 22

21/ 21

Index	Original Payload Space	Expanded Payload Space	Comments
1	00	(00,00)	Same as Original
2	01	(00,01)	New code
3	10	(00,10)	New code
4	11	(00,11)	New code
5		(01,00)	Repeated codeword
6		(01,01)	Same as Original
7		(01,10)	New code
8		(01,11)	New code
9		(10,00)	Repeated codeword
10		(10,01)	Repeated codeword
11		(10,10)	Same as Original
12		(10,11)	New code
13		(11,00)	Repeated codeword
14		(11,01)	Repeated codeword
15		(11,10)	Repeated codeword
16		(11,11)	Same as Original

FIG. 23