jc520 U.S. PTO
06/08/99

PROVISIONAL PATENT APPLICATION cover sheet

A

This is a request for filing a PROVISIONAL APPLICATION under 37 C.F.R. § 1.53(b)(2) PROV

CERTIFICATE OF EXPRESS MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Services "Express Mail Post Office to Addressee" service under 37 CFR 1.10, in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on June 8, 1999.

Express Mail Label No. **EL069171795US**

EL069171795US

Signed: Heidi L. Opstedal

Heidi L. Opstedal

	Docket No.	07451.6001-00000	Type a plus sign (+) inside this box → +

INVENTOR(S)/APPLICANT(S)

LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (City & Either State or Foreign Country)
SERRET-AVILA	JAVIER		London, United Kingdom

TITLE OF THE INVENTION ***(280 characters max.)***

METHOD AND SYSTEM FOR WATERMARKING DATA

jc541 U.S. PTO
60/138171
06/08/99

CORRESPONDENCE ADDRESS

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.
1300 I Street, N.W.
Washington, D.C. 20005
Telephone No.: (202) 408-4000

ENCLOSED APPLICATION PARTS ***(check all that apply)***

- ☒ Specification consisting of 23 pages.
- ☐ Small Entity Statement
- ☒ Drawing(s)
  - Number of Sheets: 12 sheets of drawings (Figs. 1-7B)
- ☐ Other (*specify*):

METHOD OF PAYMENT ***(check one)***

☐ A check or money order is enclosed to cover the Provisional filing fees. ☒ The Commissioner is hereby authorized to charge filing fees and credit Deposit Account Number 06-0916.	PROVISIONAL FILING FEE ☒ $150.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

- ☒ No.
- ☐ Yes, the name of the U.S. Government agency and the Government contract number are:

Respectfully submitted,

SIGNATURE: David L. Clark Date: June 8, 1999

Typed or Printed Name: David L. Clark Registration No.: 37,082

☐ Additional inventors are being named on separately numbered sheets attached hereto.

PROVISIONAL APPLICATION FILING ONLY

METHOD AND SYSTEM FOR WATERMARKING DATA

FIELD OF THE INVENTION

The present invention relates generally to systems and methods for protecting content that is stored or distributed using electronic media. More specifically, the present invention relates to systems and methods for controlling access to electronically stored information through the use of watermarking and digital signature techniques.

BACKGROUND OF THE INVENTION

As the electronic storage and transmission of music, movies, documents, and other forms of proprietary content becomes more prevalent, owners of proprietary content have grown increasingly concerned with protecting that content from unauthorized use. Accordingly, there has been growing support for a variety of standards and encoding formats designed to prevent the unauthorized use of proprietary electronic content. These standards and encoding schemes generally seek to protect proprietary content by preventing unauthorized users from being able to access or otherwise derive benefit from the content. For example, a music player that is compliant with such a standard might be unable to play unauthorized music, the unauthorized character of the music being derived from the encoding of the music itself.

However, since there is a large volume of previously-distributed electronic content and equipment that is not compliant with the new standards and encoding

IPR2020-00660 Page 00002

schemes, there is also a need for new devices and standards to be backwards compatible, as owners of out-dated equipment, or content encoded in old formats, will be reluctant to purchase new equipment or to embrace a new encoding standard that renders their existing collection of content and/or equipment obsolete or less convenient to use. Thus, there is a need for new devices and standards to support content encoded in the existing formats.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be readily understood by the following detailed description in conjunction with the accompanying drawings, wherein like reference numerals designate like structural elements, and in which:

Fig. 1 is an illustration of a system for practicing an embodiment of the present invention.

Fig. 2A and 2B are illustrations of conventional signature encoding and watermarking techniques.

Fig. 2C is an illustration of a technique for using a signature to verify the integrity of a content stream.

Fig. 2D is an illustration of a watermarking scheme in accordance with the principles of the present invention.

Fig. 3 is a block diagram of a system for watermarking a data stream.

Fig. 4 is a block diagram of a watermarking system implemented in conjunction with a compression engine.

Fig. 5A is an illustration of a watermark and signature verification engine.

Fig. 5B illustrates a technique for detecting block boundaries.

Fig. 6 illustrates a hashed signature concatenation system.

IPR2020-00660 Page 00004

Fig. 7A is a flow chart illustrating an encoding procedure in accordance with one embodiment of the present invention.

Fig. 7B is a flow chart illustrating a decoding and verification procedure in accordance with one embodiment of the present invention.

IPR2020-00660 Page 00005

DETAILED DESCRIPTION

A detailed description of a preferred embodiment of the invention is provided below. While the invention is described in conjunction with this preferred embodiment, it should be understood that the invention is not limited to any one embodiment, and encompasses instead, numerous alternatives, modifications and equivalents. For example, while the following embodiments are described in the context of a system and method for applying and utilizing watermarks and digital signatures to encode and protect real-time audio signals encoded in red book CD and minidisk formats, those skilled in the art will recognize that the disclosed systems and methods are readily adaptable for broader application. For example, without limitation, the present invention could be readily applied in the context of video, audio-visual, multimedia, or other data – both real-time and non-real-time – encoded in a variety of formats. While numerous specific details are set forth in the following description in order to provide a thorough understanding of the present invention, the present invention may be practiced without some or all of these specific details. Moreover, for the purpose of clarity, certain specifics relating to technical material that is known in the art related to the invention have not been described in order to avoid unnecessarily obscuring the present invention in such detail.

For purposes of discussion, content will generally be classified as registered or unregistered. Registered content will refer to content that is compliant with, or embodies, a predefined data protection or encoding standard or technology. Unregistered content, on the other hand, will generally be used to refer to content that is not compliant with the

IPR2020-00660 Page 00006

predefined data protection or encoding standard, and/or that does not embody the predefined data protection or encoding technology – whether as a result of operations performed on registered content, or by virtue of the fact that the content was never registered in the first place (e.g., content encoded in pre-existing formats).

The systems and methods described herein enable the secure transmission and use of high-quality signals registered according to a predefined format, and also enable the use of signals encoded in other, less secure formats. Specifically, in a preferred embodiment a hard-to-remove, easy-to-retrieve, strong watermark is inserted in the data signal. The data signal is divided into a sequence of blocks, and digital signatures of the blocks are embedded in the blocks using watermarks. When a user plays or uses the data sequence, the sequence is checked for the presence of the watermark containing the digital signature. If this watermark is found, then the digital signature is used to verify the authenticity of the content. If this watermark is not found, or the signature does not very, then the data sequence is checked for the presence of the strong watermark. If the strong watermark is found, this indicates that someone has tampered with the data sequence, and further use of the data sequence is inhibited. If, on the other hand, the strong mark is not found, then further use of the sequence is allowed. Thus, the system inhibits the use of previously-registered content that has been tampered with, but allows the use of content that was not previously registered.

Fig. 1 is an illustration of a system 100 for practicing an embodiment of the present invention. As shown in Fig. 1, system 100 includes an encoding system 102,

IPR2020-00660 Page 00007

such as computer 110; a decoding system 104, such as playing device 150; and a system 106 for communicating therebetween.

Encoding system 102 preferably includes:

- one or more input devices, such as microphone jack 112, input port 114, disk drive 115, CD-ROM drive, and/or DVD drive for receiving an analog or digital signal;
- a signal processor 116 for receiving a signal from an input device such as jack 112 and converting it to a predefined format, such as pulse-code modulated (PCM) form;
- a processing unit 118;
- system memory 120, preferably including both high speed random access memory (RAM) and read only memory (ROM), for storing system control programs, data, and application programs for e.g., applying security techniques to a data signal;
- a user interface 122, including a display 124 and one more user input devices 126;
- one or more output ports 128 for transmitting the data signal, or a processed version thereof, to one or more external devices, such as CD-writer 130 and/or network 142; and
- one or more internal buses 132 for interconnecting the aforementioned elements of the system.

The operation of system 102 is controlled primarily by programs executed by the system's processing unit 118. The system's control programs may be stored in system memory 120, and in a typical implementation include programs for, e.g., dividing a data signal into blocks, applying both weak and strong watermarks to the data signal,

IPR2020-00660 Page 00008

determining signatures for data blocks, and/or compressing the data signal. It will be appreciated, however, that some or all of these functions may be performed in conjunction with, or entirely by, appropriate hardware.

Communications system 106 can encompass any suitable system or device for transporting data from encoding system 102 to decoding system 104, including a digital or analog network 142 or the manual transportation of a magnetic or optical disk from one system to another, or any combination of these or other suitable techniques.

Decoding system 104 is operable to decode the signal encoded by system 102, to apply security transformations to the signal, and to output the decoded signal to a user, if appropriate. In one embodiment, decoding system 104 may include:

- a processing unit 152;
- system memory 153, preferably including a combination of both RAM and ROM for storing system control programs, data, and application programs for e.g., applying security techniques to a data signal;
- a non-volatile storage unit 155 containing magnetic and/or optical storage, such as a magnetic disk drive, a CD-ROM drive, a DVD drive, and/or a minidisk drive;
- an input port 157 for receiving signals from an external source, such as network 142 or external disk drive 154;
- a signal processor 156 for converting a digital signal into analog form;
- an output port 158 for sending an analog signal to an output device, such as a pair of speakers.

IPR2020-00660 Page 00009

In a preferred embodiment, a device is designed to play or use digital content that embodies a new standard or encoding technology and to resist attempts to play or use unauthorized content. To do this, the device distinguishes between content that is compliant with the standard or technology and content that is not. However, in order to provide backwards-compatible support for pre-existing or alternative formats, the device is also able to distinguish between registered content that has always been registered, registered content that was previously unregistered, unregistered content that has always been unregistered, and unregistered content that was once registered. That is, the device is able to recognize and accept pre-existing content that was never registered, and is also able to detect previously-registered content that an attacker tries to make appear as if it had never been registered. Thus, the device can generally accept data that has always been registered or has always been unregistered, while rejecting content that tries to appear registered but was previously unregistered, and content that tries to appear unregistered but was previously registered.

Detection of Unregistered Content that Attempts to Appear Registered

Detection of unregistered content that attempts to appear registered typically involves the detection of an attacker's attempt to make unauthorized or unregistered content appear as though it were authorized, thus enabling a standard-compliant device to play or use the unregistered content.

In the case of content that was at one time registered, it may have passed from being registered to being unregistered by, for example, a copying process. Such a

IPR2020-00660 Page 00010

process may include, for example, lossless bit-by-bit copying or lossy or transformed copying.

Lossless bit-by-bit copying is generally difficult to detect, since effective detection usually requires physical mechanisms that detect the devices that actually perform the copying operation, and thus typically requires modifications to be introduced into exiting hardware, such as consumer appliances. This is typically not desirable, however, as consumers typically demand backwards compatibility.

Lossy or transformed copying is typically the more common case of concern. Here, content goes through at least one lossy copying or other transformation after it has been registered. Accordingly, this type of copying can be detected by including in the original registration process a mark that is relatively difficult to introduce, but relatively easy to remove. In addition to being difficult to introduce, and easy to retrieve, such a mark also preferably is difficult to verify if any transformation has been applied to the signal.

These requirements are generally met by cryptographic signatures. However, signatures cannot be attached to real-time audio content in the conventional manner, since doing so results in a degradation of sound quality and/or security. This is illustrated in Figs. 2A through Fig. 2C, which show, *inter alia*, block diagrams of conventional signature encoding techniques. Referring to Fig. 2A, a stream of encoded data 200 is shown. Data 200 could represent, for example, a stream of PCM data representing an

IPR2020-00660 Page 00011

audio track from a compact disk. There are a number of ways to attach a signature to data 200.

One simple technique is simply to sign the entire stream of data with a single, long signature. This approach is typically undesirable, however, as a signature is generally very sensitive to errors or variations in the data, as such variations cause the signature to fail. Since the PCM data signal 200 typically will contain relatively short, random burst errors, use of a signature that is too long is prone to failure due to these errors, as opposed to the errors or modifications made by an attacker, which is the type of modification that the signature should detect.

An alternative approach, then, is to break data stream 200 into a stream of data blocks 204. Each block 206 having its own associated signature 208. The size of the blocks 206 can be made small enough to minimize the likelihood that the random burst errors present in the PCM signal will be present in a predetermined fraction of the data blocks 206, and large enough to ensure that the signature 208 associated with each block is not easy to crack.

The problem with the approach shown in Fig. 2A, however, is that when signatures 208 are inserted into data stream 204 (for example, in the manner shown in Fig. 2B), they can produce undesirable sounds when data stream 204 is played. Since audio quality is typically the primary concern of a consumer using an audio player, this type of degradation of the audio signal embodied in stream 200 is unacceptable.

IPR2020-00660 Page 00012

Reducing the size of the signature to avoid audible detection, while maintaining a high level of security, is usually not possible.

Another technique might be to transparently embed a signature in each data block using a watermark (210). By embedding the signal in this manner, unacceptable degradation of the audio signal could be substantially avoided. However, a problem with this approach is that the very process of embedding the signature in the data is likely to change the data slightly, thus rendering the signature unable to retrieve the original data. Thus, the system designer is faced with a catch-22 situation: the signature is only valid if it is not inserted into the data, yet it must be inserted into the data in order to be of any use.

Figs. 2D and 3 illustrate an embodiment of the present invention that solves these problems by embedding a block's signature in the watermark of the block that follows the block to which the signature corresponds. Referring to Fig. 3, a PCM signal 302 is received by a mark-injection engine 300, which is operable to insert a watermark containing a digital signature. Incoming PCM data is stored in an input buffer 303. When a predetermined amount of data has accumulated in input buffer 303, the data is sent to watermarking engine 304, which inserts a watermark into the PCM data 302 to yield watermarked PCM data 306. Watermarked PCM data 306 is then sent to a user, while a copy is sent to signature engine 308. Signature engine 308 is operable to create a signature 310 that corresponds to watermarked PCM data 306. Signature 310 is then input to a delay element 312. Delay element 312 is operable to delay signature 310 until the next block of input PCM data 301 is ready to be input to watermarking engine 304, at

IPR2020-00660 Page 00013

which time signature 310 will be output from delay element 312 and input to watermarking engine 304. Thus, signature 312 of the watermarked version of input data block 302 is included in the watermark of data block 301.

This process is repeated for each data block in the data stream, the result being a data stream in which each block, except for the first and the last, is watermarked with the signature of the block just ahead of it in the transmission stream. The first data block that is transmitted will typically not contain a signature, and the last block that is transmitted will not have a signature that corresponds to it, since there is no block that comes after it. Thus, the present invention is advantageously able to obtain the high level of security provided by digital signatures, while benefiting from the transparency of watermarks, and is thus able to transmit data securely without suffering from undesirable degradation in the quality of the data stream.

Since a signature will be corrupted if a transformation is applied to the signal, a work-around must be found in order to support lossy-compressed, registered material (e.g., MD). As shown in Fig. 4, the present invention solves this problem by modifying the technique discussed above with reference to Fig. 3. Specifically, the present invention advantageously enables the watermarking/signature technique, described above, to be applied in conjunction with either lossy or lossless compression. Referring to Fig. 4, PCM data 402 is input to mark injection engine 400. Mark injection engine 400 includes a watermarking engine 404 for receiving input PCM data 402 and inserting a watermark to form watermarked PCM data 406. Watermarked PCM data 406 is then sent to a compression engine 408 which compresses the data. Compressed data 410 is sent to

IPR2020-00660 Page 00014

a user or to another device for further processing, while a copy of compressed data 410 is input to decompression engine 412. Decompression engine 412 reverses the compression process, yielding decompressed PCM data 414. Decompression engine 412 preferably is chosen to emulate realistically the decompression that will be employed by users of the data. If the compression performed by compression engine 408 (and the decompression performed by engine 412) is lossless, then decompressed data 414 will be the same as watermarked PCM data 406. However, if lossy compression is used, then decompressed data 414 will not be expected to be the same as watermarked PCM data 406.

Decompressed data 414 is input to signature engine 416 which derives an electronic signature 418 corresponding to the data 414. This signature is then sent to a delay block, where it waits until the next block input PCM data 401 is ready to be watermarked by engine 404, at which time it is input to watermarking engine 404. As one of ordinary skill in the art will appreciated, one or more buffers (not shown) can also be inserted between the various blocks of Fig. 4 in order to ensure the proper timing of the data and signal flow through mark injection engine 400.

Accordingly, mark injection engine 400, like mark injection engine 300, is able to advantageously combine the use of digital signatures with the use of watermarks, thus achieving a high level of security while maintaining the quality of the input data signal. Moreover, engine 400 is able to achieve these goals even if lossy compression is applied to the input signal. Specifically, by decompressing the compressed data 410 before obtaining signature 418, engine 400 ensures that signature 418 will be the appropriate signature for the data block 414 that a user will obtain after decompressing block 410.

IPR2020-00660 Page 00015

It should be understood that mark injection engines 300 and 400 can be implemented as any suitable combination of hardware and/or software components. For example, in a preferred embodiment, a conventional watermarking engine is used in conjunction with a conventional signing engine and delay elements. Moreover, any suitable compression engine can be used in connection with engine 400. In addition, it should be noted that both the process shown in Fig. 3 and the process shown in Fig. 4 produce an encoded data sequence that can be verified by a single decoding test engine.

A block diagram of the process performed by a decoding device, such as audio player 150, is shown in Fig. 5A. As shown in Fig. 5A, a decoding device 502 is configured to decode an input stream of data 504 and to either inhibit or allow use of the data depending on the results of the decoding process. In one embodiment, decoding device 502 is able to handle input data in either compressed or uncompressed form. As shown in Fig. 5A, if input data 504 is in compressed form, it is first preferably passed to decompression engine 506 which decompresses the data into an uncompressed signal of, e.g., PCM data. Once a stream of PCM data has been obtained, either directly or from decompression engine 506, it is passed to mark verification engine 508. Mark verification engine 508 is operable to receive blocks of PCM data, detect the presence of watermarks and signatures in those blocks, and to extract the signatures if they are present. Since the signature for a given block, A, is embedded in the watermark for the following block, $A+1$, incoming blocks are held until the next block has been received and the signature has been extracted from it. The signature, and the block to which it corresponds, are then processed by signature verification engine 510, which verifies the

IPR2020-00660 Page 00016

integrity of the block using the signature. In order to make re-processed content (e.g., illicit unregistered content) pass this verification test, it will generally be necessary to reproduce the whole signal, which is typically impractical.

In order to process the watermarked/signed data in the manner shown in Fig. 5A, a decoding engine needs to be able to detect the block boundaries so that it can detect and process the corresponding watermarks and signatures. However, even though watermarking algorithms typically can auto-synchronize themselves, the task of detecting block boundaries is made difficult by the fact that the PCM data stream does not typically include synchronization information apart from the fact that it starts on a double byte boundary. Thus, block synchronization information generally cannot be extracted directly from the PCM data stream.

To overcome the lack of synchronization in the pure PCM signal, one embodiment of the present invention introduces a "guess" into the information contained in the watermark, the guess being operable to enable the signature-verifying engine to find the signed block more easily. As the guess typically cannot contain position information, it contains an easy-to-compute representative value of the block.

In one embodiment, the size of the representative value is chosen to be much smaller than the watermark block size. However, to enhance security the size of the signed block is chosen so that it is clearly audible, and the rate of signed blocks is chosen so that it is subjectively perceptibly high. Not all the blocks are signed, just one per watermark block, and the signed block is localized in the middle of the watermark block

IPR2020-00660 Page 00017

This technique is illustrated in Fig. 5B. The guess contains an easy-to-compute, representative value that allows the signature engine to localize the block with a low failure rate (false positives). This optimization allows the verifier system to avoid checking all the windows of signature-bloc-size in the watermarking block for a possible match. Since this guess typically contains less information about the block than the signature itself, it generally does not provide additional security, which is why it is not signed as well. In a preferred embodiment, a simple double-word-sized logical exclusive-or (XOR) is used as the guess, as it is easy to compute on the fly and meets all the requirements. However, it will be appreciated that other techniques for implementing the guess could be used, where, in general, such techniques are characterized by the relation:

$$Block\text{-}A\ [TRANSFORM]\ Block\text{-}B = X;\ and$$

$$Block\text{-}A\ [TRANSFORM']\ X = Block\text{-}B$$

It is seen that XOR can advantageously be used for both [TRANSFORM] and [TRANSFORM'].

Several other technological constraints introduced by the environment and the available technology will typically apply to implementations of the process described above. For example:

- The PCM signal, extracted from, e.g., a CD, will generally not be completely reliable;
- Decompression engines are generally not 100% deterministic;

IPR2020-00660 Page 00018

- The signing key must be robust against attacks; and
- The watermarking algorithm typically cannot transport a large amount of data.

In order to overcome these problems without undesirably compromising security, several measures can be taken. For example, with regard to the reliability of the extracted PCM signal, experiments show that the signals from real devices suffer from errors that conform to a burst error model. What this tends to indicate is that only a small percentage of the signed blocks are actually affected by errors. However, since conventional digital signatures are typically of an all-or-nothing nature, even a failure in a small percentage of the signature can lead to the mistaken rejection of registered content and/or to signal degradation.

The present invention solves this problem by using a threshold for signature acceptance, based on the percentage of good blocks detected. Only those signals that contain a minimum absolute number of good blocks per unit are accepted. For example, in one embodiment a group of blocks are accepted only if at least 80% of the blocks are valid, regardless of whether errors have corrupted the signature in the remaining 20% of the blocks.

With regard to the fact that decompressing engines are typically not completely deterministic, it has been observed that the lower significant bits of the signal are effectively randomly inserted by the decoder. In order to account for this, in one embodiment the watermarked signal includes (along with the signature and the guess), a two-bit field with information on the reliability of the most insignificant bits of the signal.

IPR2020-00660 Page 00019

In other words, the two-bit signal effectively indicates how sample bits should be included in the signature. Bits not included in the signature can be assumed to be zero. An illustrative encoding of this two-bit signal is:

- 00: All 16 bits are relevant (to be used with, e.g., red-book CDs);
- 01: Only the 12 most significant bits are relevant;
- 10: Only the 10 most significant bits are relevant;
- 11: Only the 8 most significant bits are relevant.

One of ordinary skill in the art will appreciate that the number of bits appropriate for a particular encoding algorithm can be readily determined empirically in accordance with the principles of the present invention. This value need not be signed along with the signal because it is generally not possible to mount an attack by changing it, as the signature will not verify anymore.

Regarding the robustness of the signing key, it will be appreciated that since knowledge of the signing key is enough to produce registered material, it is desirable to protect it against all kinds of attacks. Physical attacks can generally be avoided by placing the key in a single protected environment, for example, at the content certification authority installations. To protect against cryptographic attacks, a well-proven, highly resistant public key technology can be used. For example, in one embodiment, an RSA algorithm is used with a relatively large key size (e.g., between 2048 and 4096 bits), which will produce a durable and robust key.

IPR2020-00660 Page 00020

Another technological constraint on the use of the techniques described above, is that a watermarking algorithm typically cannot transport relatively large amounts of data. In this regard, it should be noted that if all of the suggestions set forth above were adopted, the watermark would contain the following data: a two-bit quality indicator, a four-byte guess, and a 2048-bit (256-byte) signature. Thus, in this example, the watermark contains almost 261 bytes, which with current technology is a relatively large amount of data for a watermarking algorithm.

In one embodiment of the present invention this problem is solved through the use of an error-recoverable shared signature scheme, which, as described below, is resistant to errors in the signed data, and yet is generally as robust as a "normal" scheme. This technique is illustrated in Fig. 6. With reference to Fig. 6, a signed data stream 602 is first partitioned. This scheme works by hashing apart those partitions and concatenating the resulting hashes. The hash concatenation is signed, instead of padding a single hash. Since secure hashes generally behave as random data, this solution is not less secure than the standard padding techniques. If an error appears on the data, the signature will verify for all partitions except for the one that is affected. This can be readily detected and recovered from. The appropriate number of correct blocks to obtain in order to decide that the signature is correct can be readily determined using statistical analysis of the PCM signal quality.

Since all the signed blocks are spread substantially equally, it is typically only necessary to find one such block in order to localize the rest. However, the guess field must be improved as failure to find the first block will lead to failure to find the rest with

IPR2020-00660 Page 00021

the result of a total signature verification failure. To recover from this, a guess for more than one block will be included; its exact number is to be determined by empiric experiments on signal quality. The number of blocks to be included into one signature depends on the final key size and the hashing algorithm used. As an example, with SHA1 or RIPEMD160, and 2048 bits for the key size, up to 16 blocks can be included.

Other solutions to the problem of having a large watermark include making the watermarking block bigger, and/or simply making the size of the watermark smaller; however, both of these techniques tend to reduce the security of the watermarking scheme.

Detection of Unregistered Content Attempting to Appear as if Never Registered

The detection of unregistered content attempting to appear as though it had never been registered, involves detecting an attacker's attempt to make previously-registered content appear as if it were pre-existing content, so as to hide the fact that the previously-registered content is being used without authorization.

One way to achieve this goal is by using a strong watermark. Accordingly, in a preferred embodiment a hard to remove, easy to retrieve, low-bit-rate watermark is chosen. In this case, encoding a single bit into the signal in such a way that it cannot be removed will typically suffice. This watermark is preferably applied to registered content before the weak signature mark is introduced. Thus, if an attacker is able to successfully remove the weak watermark and signature, the strong watermark will remain, and will serve as an indication that the data has been tampered with. Since the watermark itself

IPR2020-00660 Page 00022

need not contain any affirmative information, it will typically be difficult for an attacker to detect or remove.

In an illustrative embodiment, this technique for using a strong watermark is combined with the techniques for using digital signatures, described above. The operation of the resulting system is illustrated in Figs. 7A and 7B. Referring to Fig. 7A, an input PCM signal 702 is received by mark-insertion engine 700. Mark insertion engine 700 first inserts a strong watermark into signal 702 (704). Next, the incoming signal 702 is parsed into *N* blocks (706), and a relatively weak watermark is embedded in each block (708), the watermark containing the signature of the preceding watermarked block, a guess regarding block boundaries, and, if compression is being used, a short indication of the number of relevant bits in the signature. After the watermark is inserted, the signature of the block is determined (710), for insertion in the next block.

Fig. 7B illustrates the operation of a decoder/player upon receipt of a signal that has been processed in the manner shown in Fig. 7A. Referring to Fig. 7B, an encoded PCM signal 714 is received by decoder/player 720. The blocks of the incoming signal are first checked for the presence of a weak watermark containing a digital signature (722), using the guess (if any) contained in the signal. If the weak watermark is not found (a "no" exit from block 724), then the input signal is searched for the presence of the strong mark (725). If the strong mark is not found (a "no" exit from block 727), then the signal is accepted, as the signal is likely to correspond to content that was never registered (e.g., preexisting music files, or legacy software). It will be appreciated, however, that any response could be chosen upon detection of this type of content. If the

IPR2020-00660 Page 00023

strong mark is found, however, then appropriate defensive action is taken (729) – for example, inhibiting further use of the signal and/or playing invalid data – as the presence of the strong watermark indicates that the content was registered at one point, but now has been corrupted or otherwise modified.

If the weak watermark is found (a "yes" exit from block 724), the signature is extracted from the watermark (726). The signature is verified (728) using, for example, the registration authority public key, which is preferably embedded in the decoder/player. If the signature is determined to be authentic, then the block can be played and processing continues with the next block of the signal. However, if the signature is not authentic, then the process can check for the presence of the strong mark (or, in other embodiments, take appropriate defensive action).

Accordingly, a system and method have been described that allow a signal to be securely encoded to substantially prevent unauthorized use of the signal's content. While attempts to remove the secure encoding can be detected and rendered ineffective, attempts to use data that was never securely encoded in this manner can be detected and allowed. Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be made, as there are many alternative ways of implementing both the process and apparatus of the present invention. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein.

IPR2020-00660 Page 00024

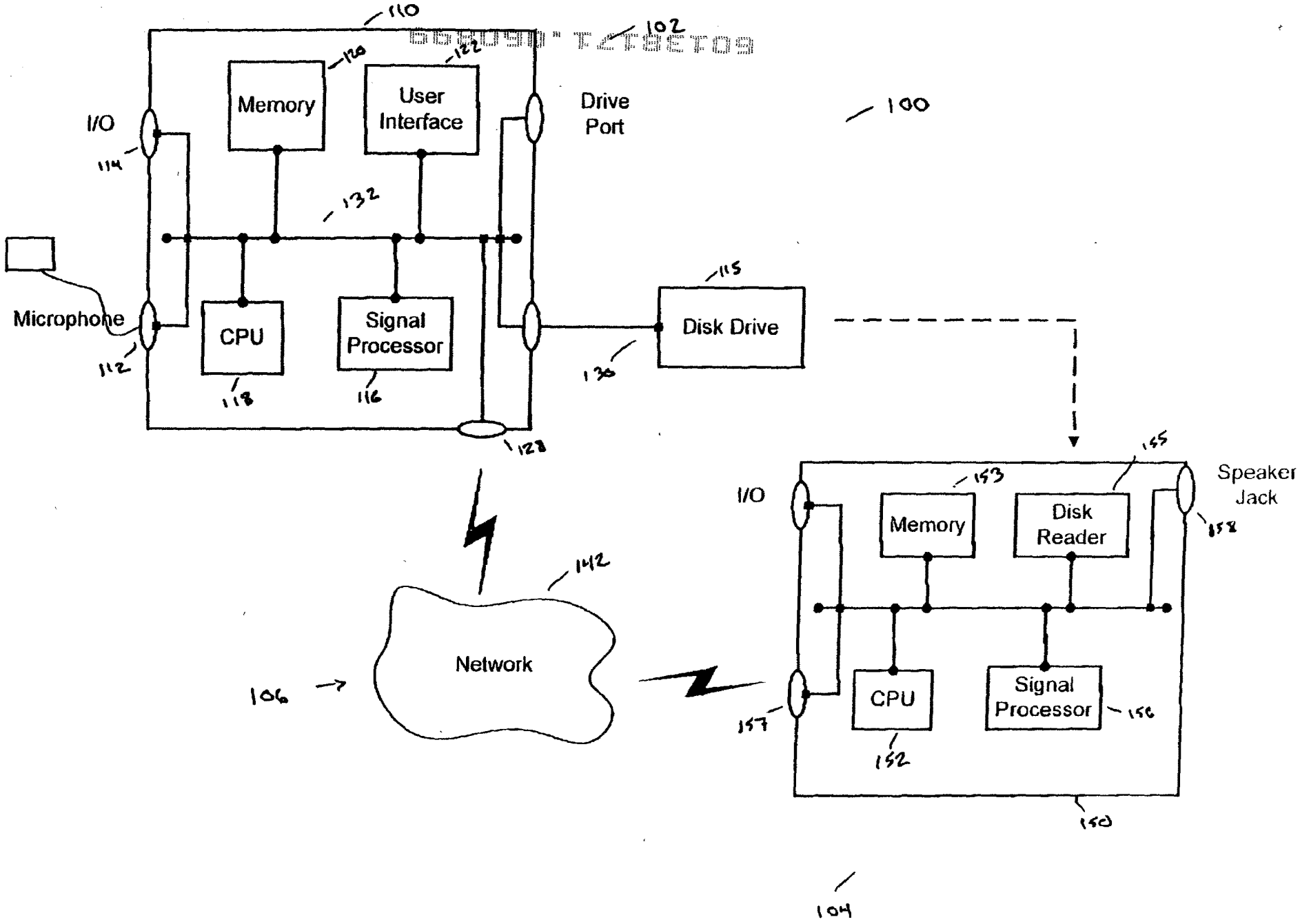

Fig. 1

IPR2020-00660 Page 00025

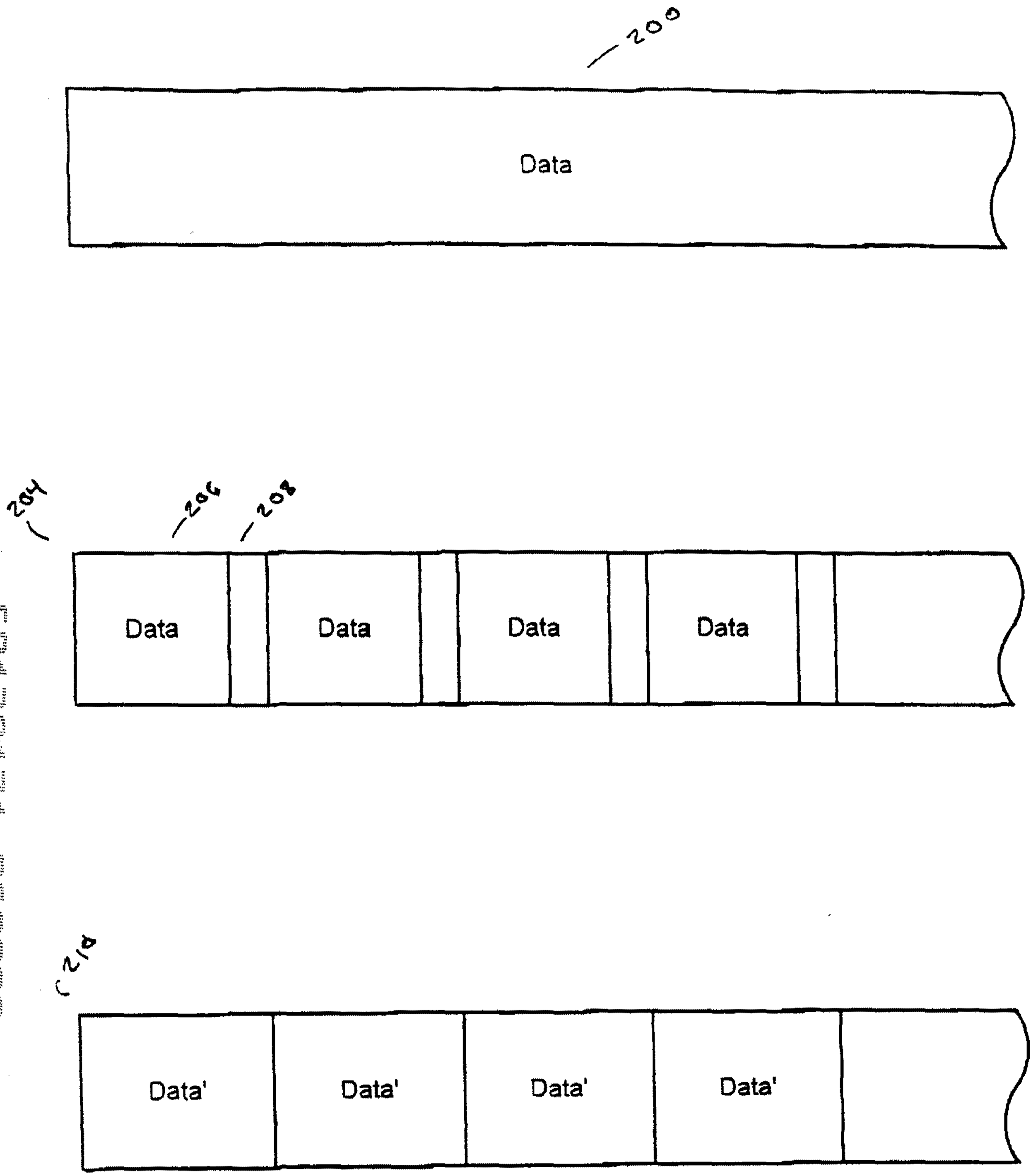
200
Data
204
206
208
Data
Data
Data
Data
210
Data'
Data'
Data'
Data'

FIG. 2A

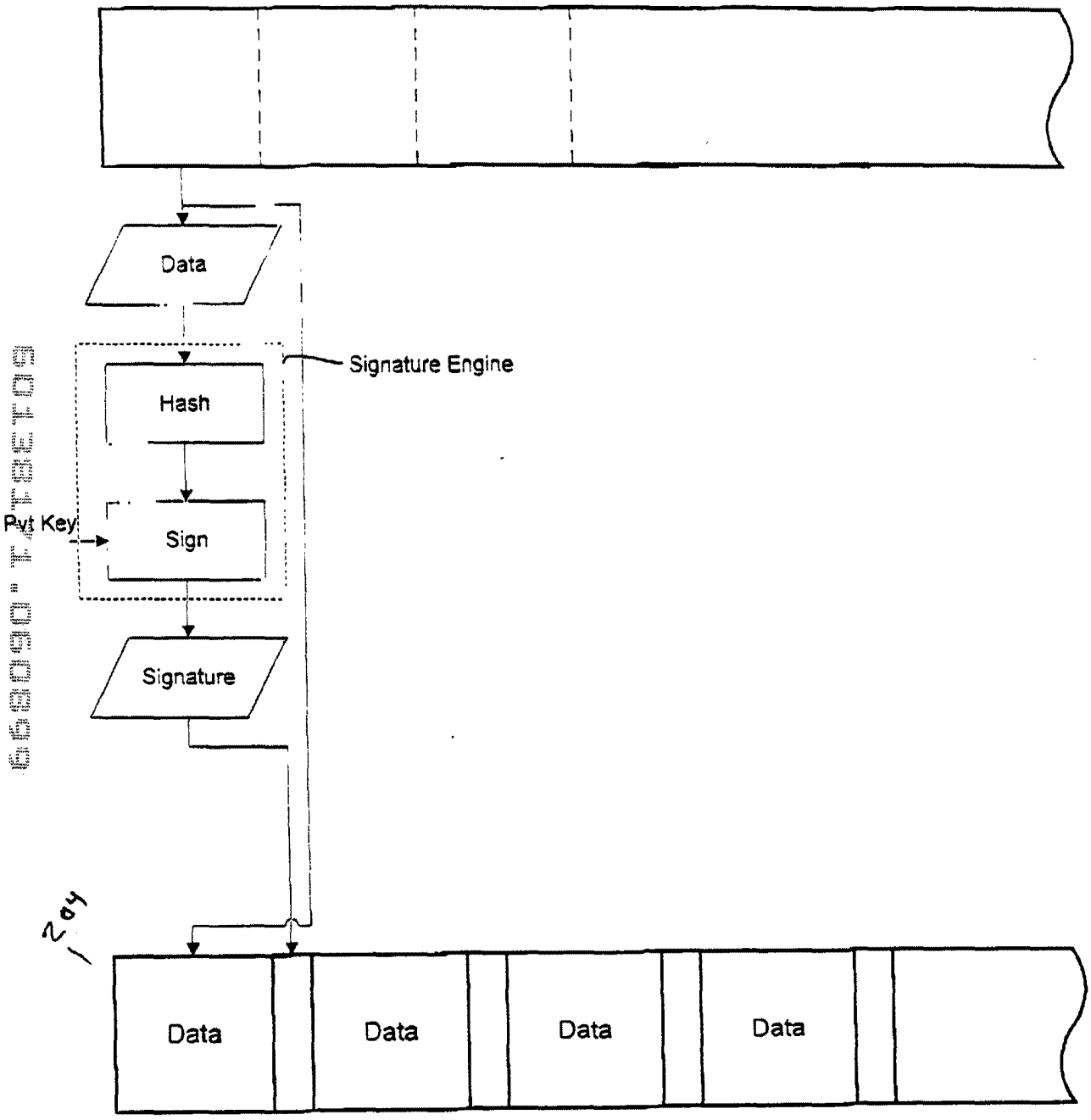
Data
Signature Engine
Hash
Pvt Key
Sign
Signature
204
Data
Data
Data
Data

FIG. 2B

6,658,568 B1 Dec. 2, 2003

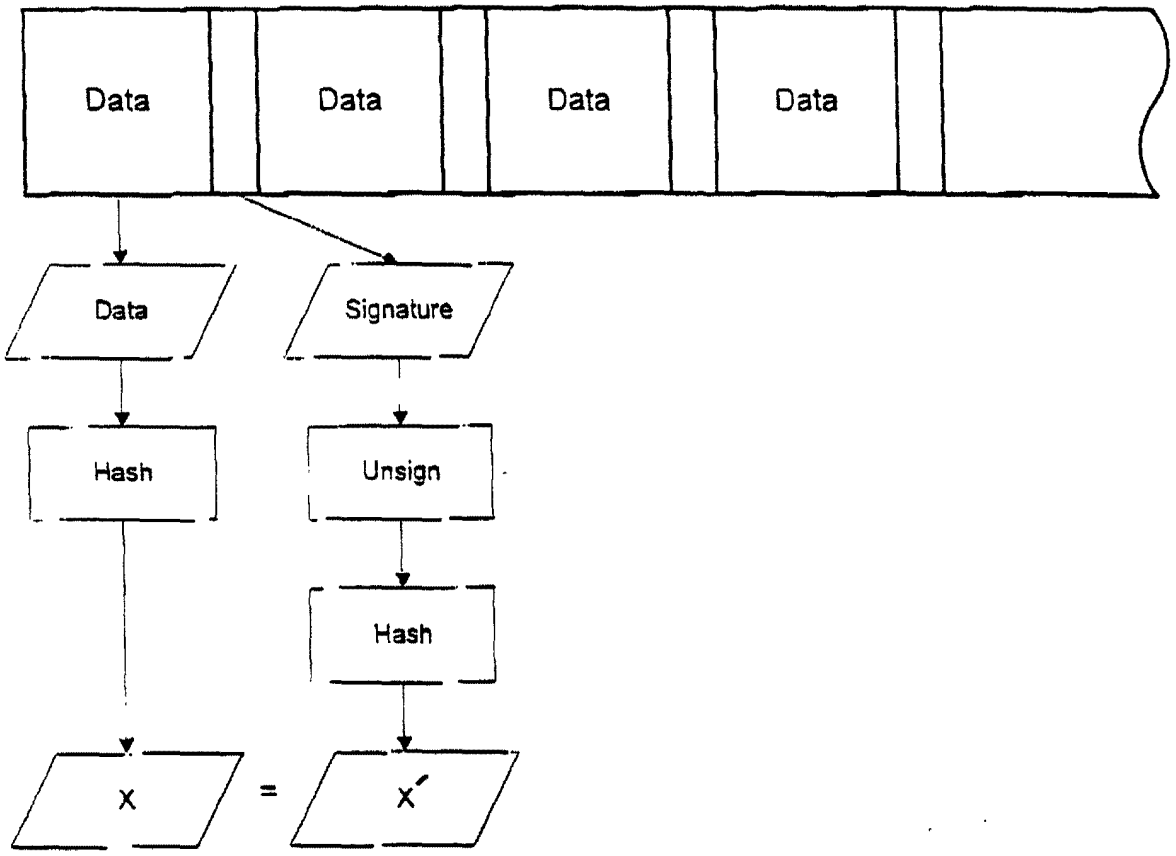

FIG. 2C

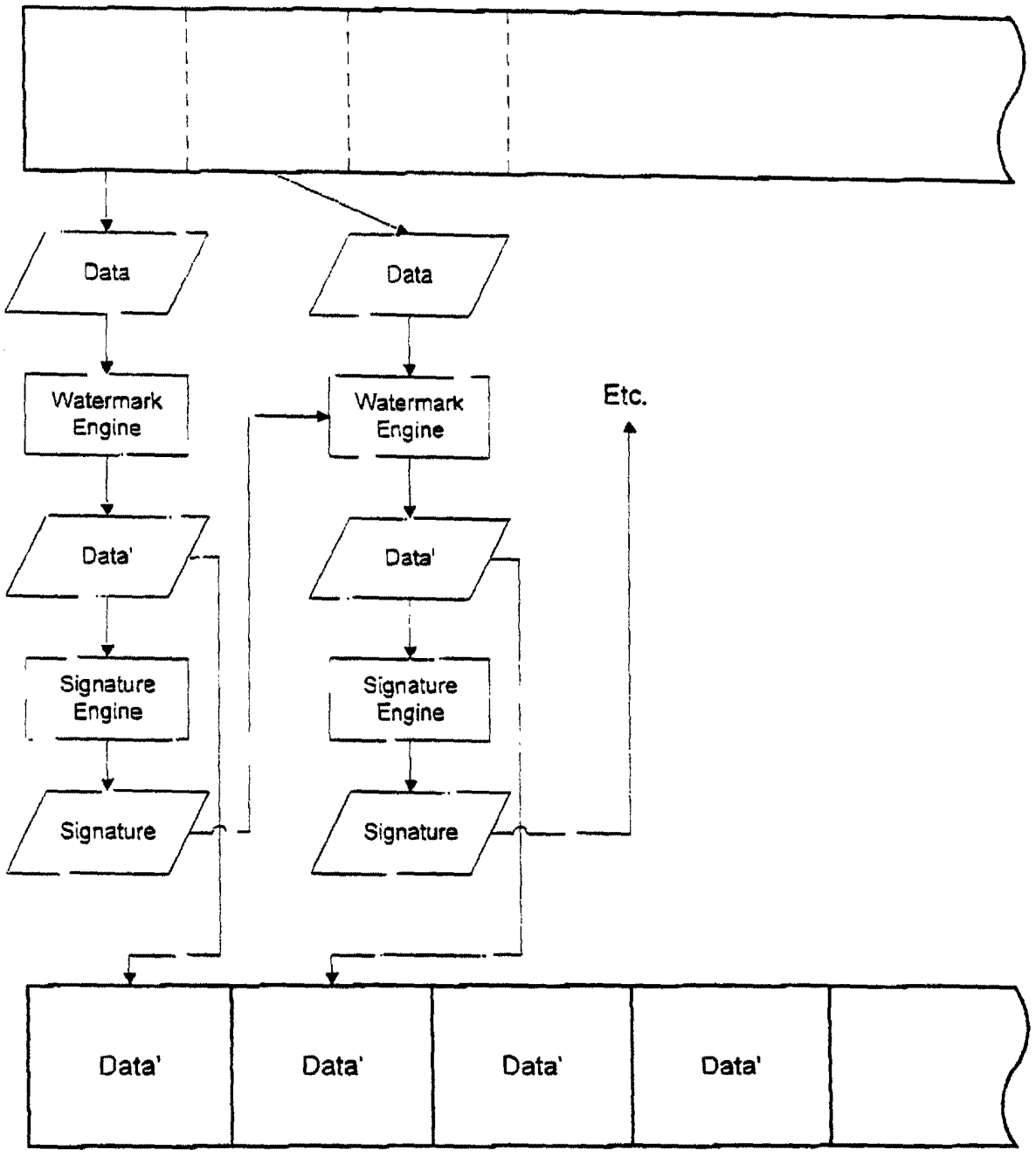
Data
Data
Watermark Engine
Watermark Engine
Etc.
Data'
Data'
Signature Engine
Signature Engine
Signature
Signature
Data'
Data'
Data'
Data'

FIG. 2D

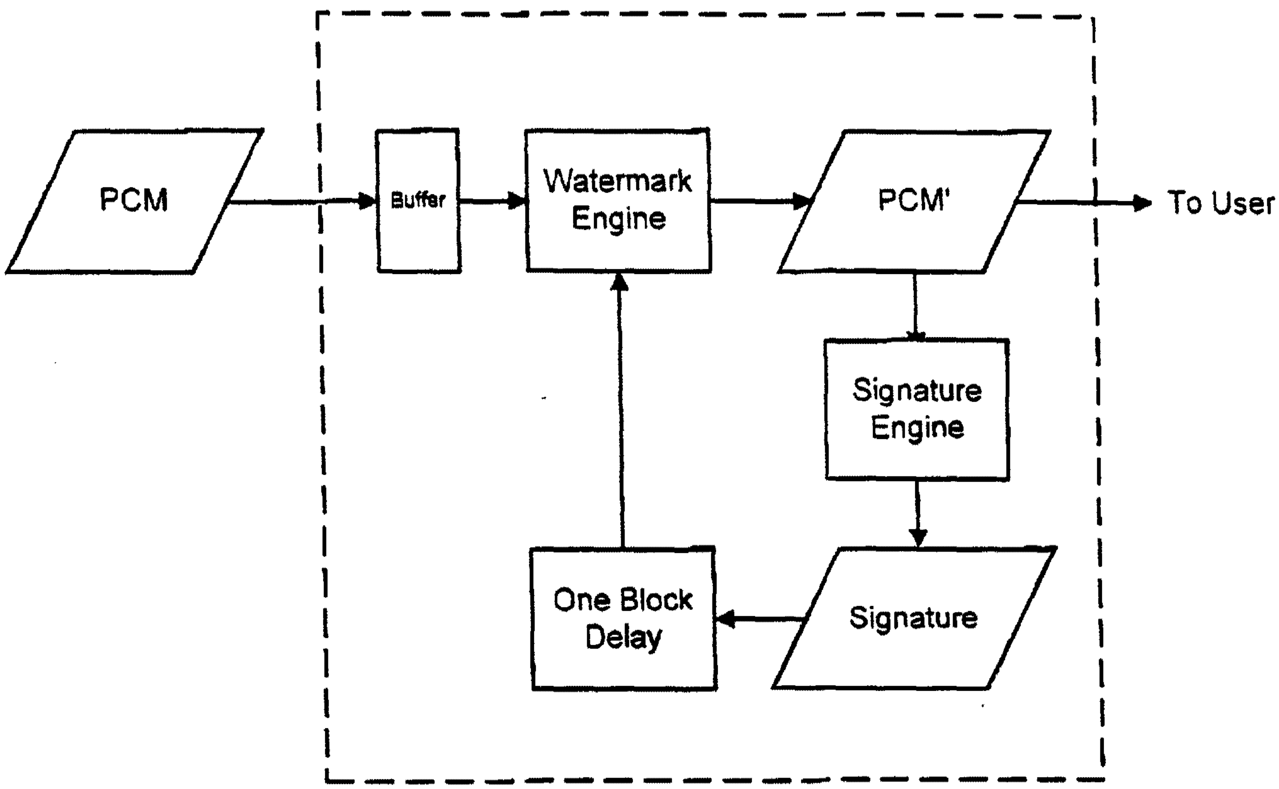
PCM
Buffer
Watermark Engine
PCM'
To User
Signature Engine
One Block Delay
Signature

FIG. 3

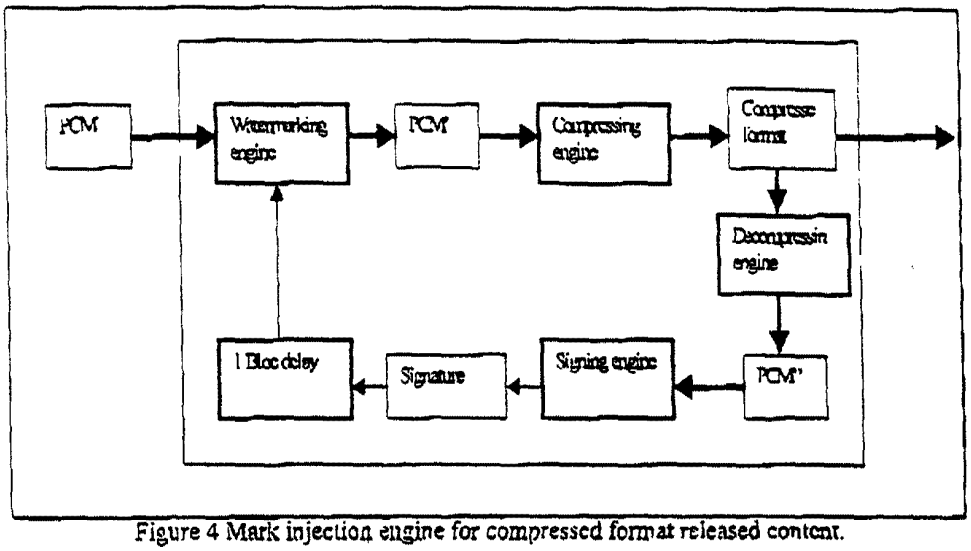

Figure 4 Mark injection engine for compressed format released content.

EP 1 904 954 B1

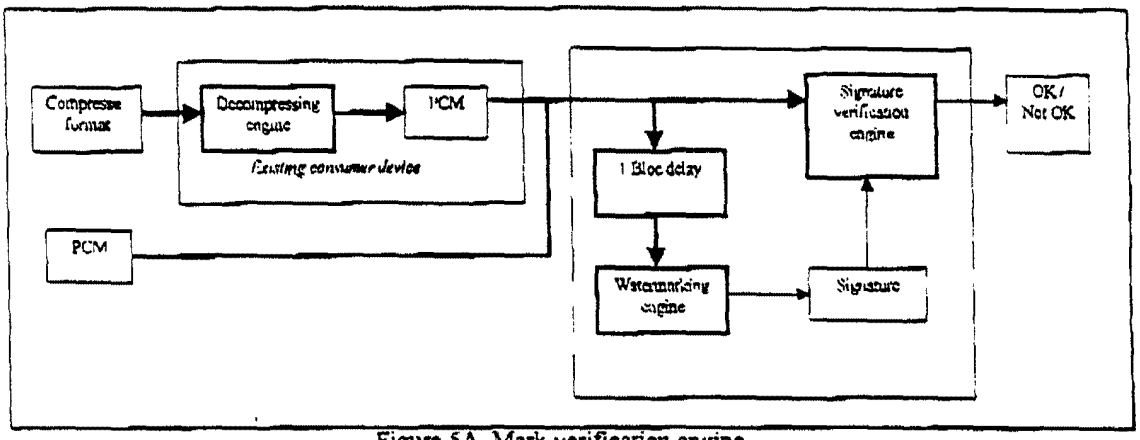

Figure 5A Mark verification engine.

US 6,598,162 B1

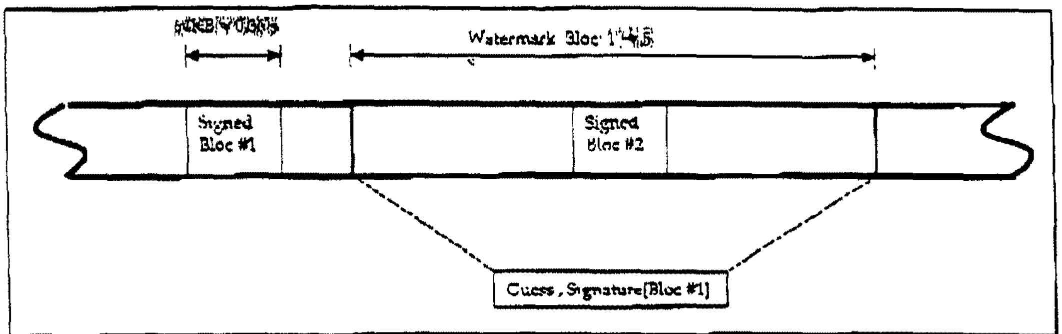

Figure 5B, Signal bloc scheme.

Sagemcom Exhibit 1006

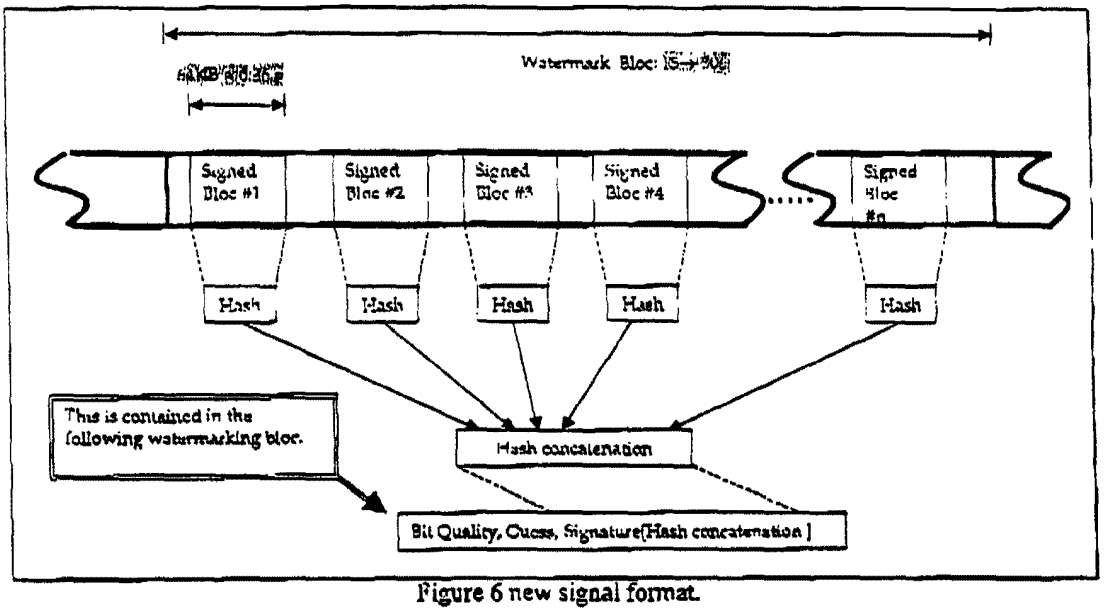

Figure 6 new signal format.

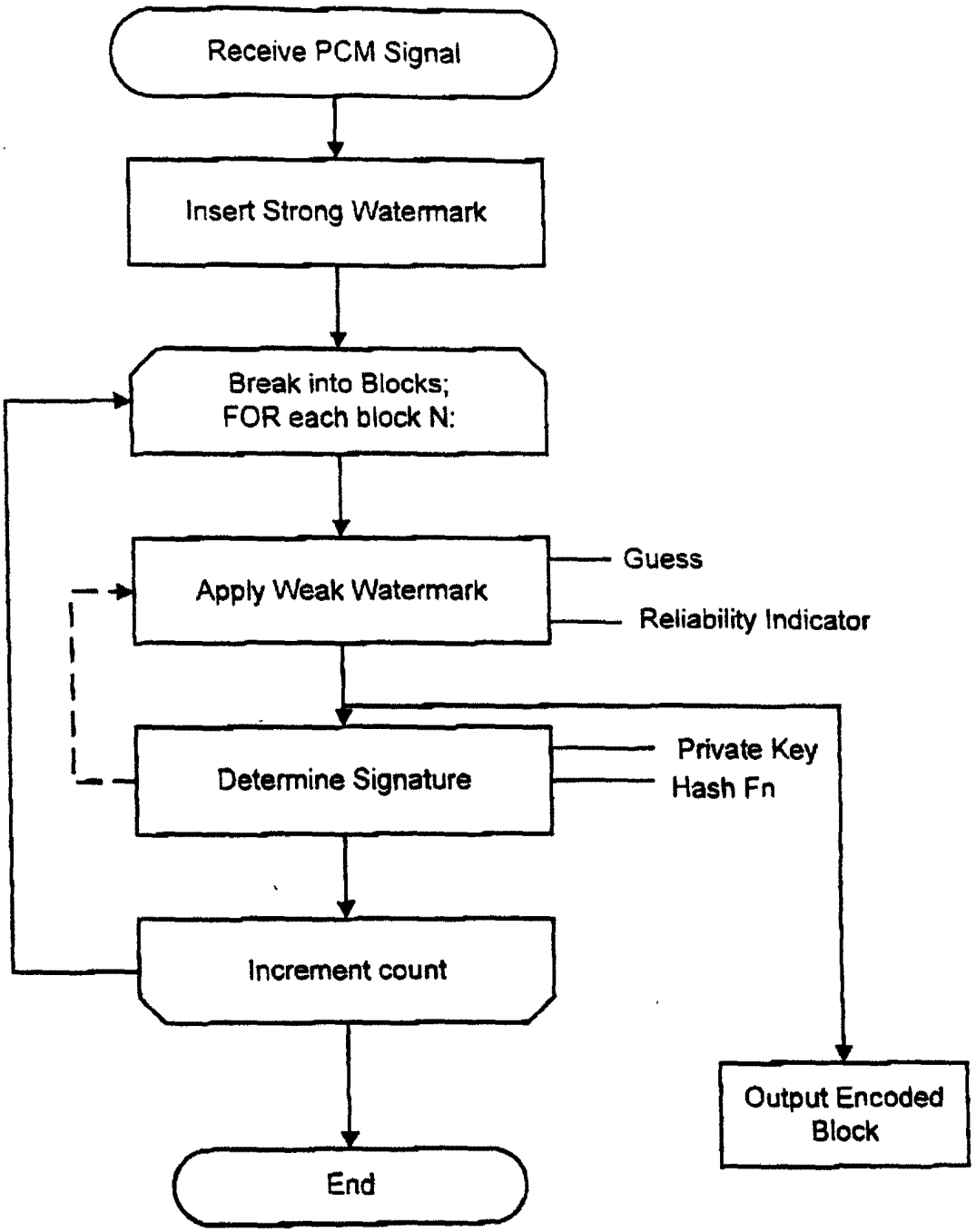

FIG. 7A

6,680,959 - 1/13/2004

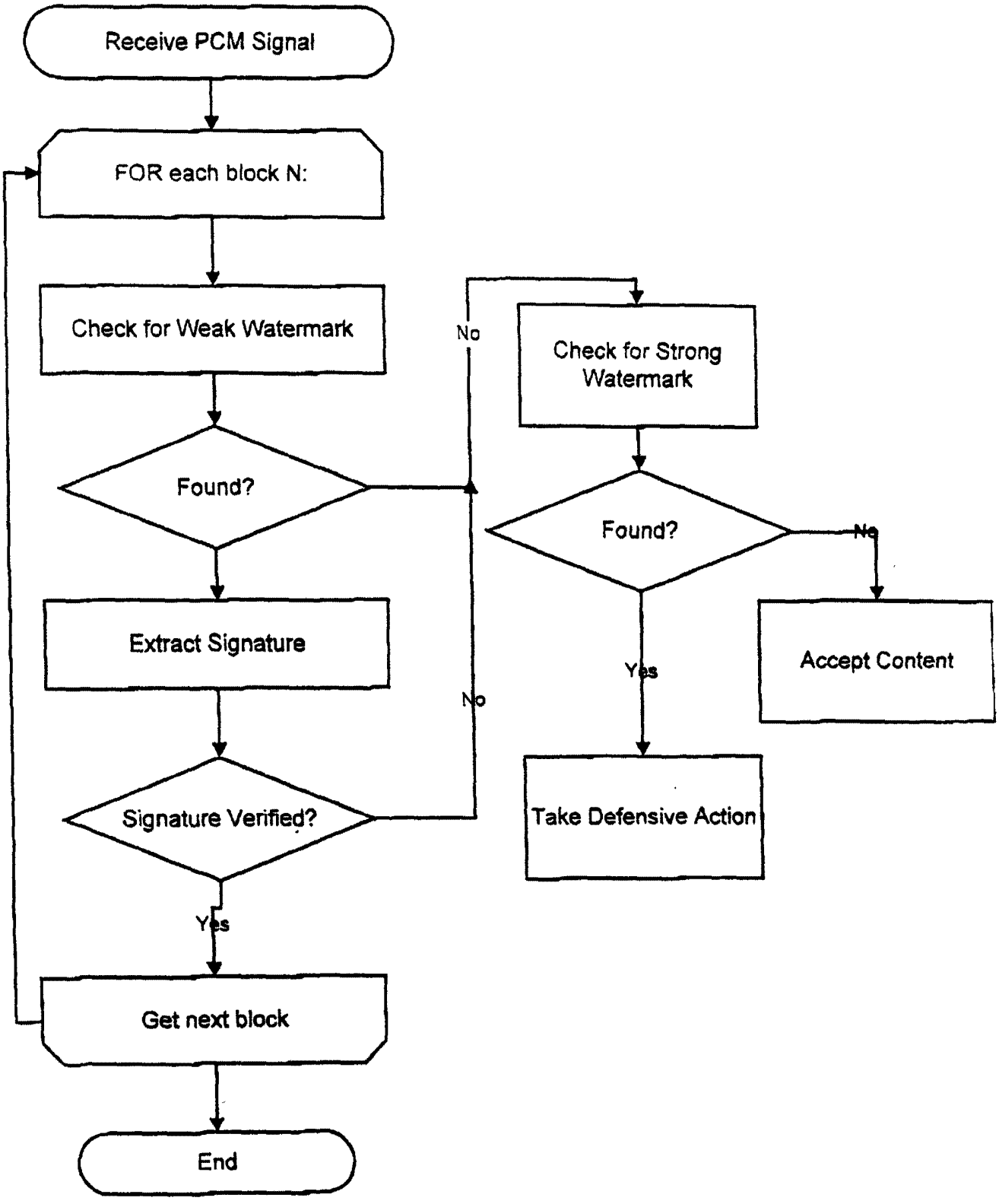

FIG. 7B

Sonos, Inc. Exhibit 1009