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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 April 2003 (10.04.2003)

PCT

(10) International Publication Number
WO 03/030541 A2

(51) International Patent Classification[7]: H04N 7/16

(21) International Application Number: PCT/US02/31488

(22) International Filing Date: 3 October 2002 (03.10.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/326,709 3 October 2001 (03.10.2001) US

(71) Applicant *(for all designated States except US)*: **THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK** [US/US]; 116th Street and Broadway, New York, NY 10027 (US).

(72) Inventors; and

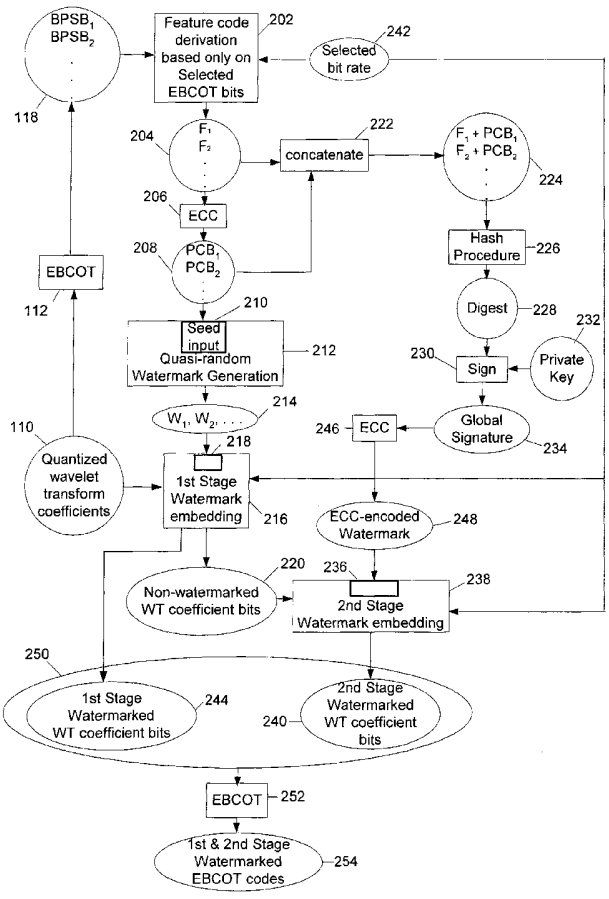
(75) Inventors/Applicants *(for US only)*: **SUN, Qibin** [CN/SG]; Block 52, #04-586, Teban Gardens Road, Singapore 600052 (SG). **CHANG, Shih-Fu** [CN/US]; 560 Riverside Drive, Apt. 18K, New York, NY 10027 (US). **MAENO, Kurato** [JP/JP]; 141-1 Kamisano-cho, Apt. #1-106, Takasaki-shi, Gunma 370-0857 (JP). **SUTO, Masayuki** [JP/JP]; 1714-1 Shinmathi, Tano-gun, Gunma 370-1301 (JP).

(74) Agents: **TANG, Henry** et al.; Baker Botts L.L.P., 30 Rockefeller Plaza, New York, NY 10112-4498 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG,

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR WATERMARKING AND AUTHENTICATING DATA



(57) Abstract: A method and apparatus for watermarking and authenticating data. Features codes are derived from multimedia data (e.g., image data) and used to derive a quasi-random watermark and an encrypted signature. In two stages of watermarking, the quasi-random watermark and the signature are embedded in the multimedia data to derive watermarked data. In an authentication procedure, the watermarked data are authenticated by newly deriving the feature codes, extracting the watermarks, and determining whether the extracted watermarks correspond to the newly derived feature codes. In addition, the embedded signature is extracted from the data and decrypted. The newly derived feature codes are used to derive data which is compared to the decrypted signature. The user of the above procedures can select a desired authentication strength based upon the percentage of the multimedia data used to derive the feature codes.

WO 03/030541 A2

SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (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*

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

METHOD AND APPARATUS FOR WATERMARKING AND AUTHENTICATING DATA

SPECIFICATION

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to U.S. Provisional Application No. 60/326,709 entitled "System and Method For Authentic Multimedia Content Within An Unified Rate-Distortion Measurement Framework," filed on October 3, 2001, which is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

Multimedia data such as digital images, video, and audio tend to be susceptible to tampering by unauthorized parties. For example, an image of a scene can be altered to add objects which were not actually present in the original image and/or to remove objects which were actually present in the original image. Not only can altered data be used for fraudulent or deceptive purposes, but one wishing to deny or misrepresent facts evidenced by genuine, unaltered data may be able to do so convincingly by clandestine, fraudulent manipulations of the genuine data because of the difficulty of detecting such manipulations.

A number of techniques have been used to detect improper data manipulation and/or to prevent repudiation of content. For example, Fig. 6A illustrates a typical prior art watermarking-based system in which original content 604 —'e.g., a set of image-domain data — is subjected to a spatial transform 608 to produce spatially-transformed data 630. Typically the spatial transform 608 is a discrete cosine transform (DCT) or a wavelet transform. Information 606 to be embedded in the content 604 is also transformed using a spatial transform 610 to derive spatially-transformed data 632. Similarly to transform 608, transform 610 is typically a DCT or wavelet transform. A watermark embedding operation 612 uses a secret key code 602 to embed data 632 into spatially transformed data 630, thereby generating watermarked data 638. Various watermark embedding operations are well known to those skilled in the art. The watermarked data 638 are then inverse

transformed (block 614) to derive a watermarked version 616 of the original content 604. For example, if the original content 604 is a set of image-domain data, the watermarked content 616 is typically also a set of image-domain data.

To authenticate content which has purportedly been watermarked using the system illustrated in Fig. 6A, the watermark is extracted and verified using the verification system illustrated in Fig. 6B. In the illustrated verification system, both the content 626 to be authenticated and a watermarked version 628 of that content 626 are spatially transformed by blocks 618 and 620, respectively, to derive spatially transformed data 634 and 636, respectively. Using the same secret key code 602 with which data 628 were purportedly originally watermarked in the procedure illustrated in Fig. 6A, a watermark extraction procedure 622 processes the spatially transformed original and watermarked data 634 and 636, respectively, to derive extracted watermark data 640. Various watermark extraction procedures 622 are well known to those skilled in the art and, for that reason, need not be further described. The extracted watermark data 640 are verified (block 624) by comparing such data to the embedding information 606 which was embedded in the original content 604 in the watermarking procedure illustrated in Fig. 6A.

However, the above-described watermarking and authentication systems can fail to detect certain types of fraudulent manipulation. For example, small objects can be added or deleted without detection. In addition, the watermarking and authentication systems illustrated in Figs. 6A and 6B, respectively, use the same secret key 602 in both the watermarking system and the authentication system. Therefore, the secret key 602 must be distributed to any party who will be authenticating data, a requirement that increases the risk that the secrecy of the key 602 will be compromised. If the key 602 is obtained by an unauthorized party, such a party can extract the watermark 640, improperly modify the content 626, re-embed the watermark, and then pass off the altered content as being genuine.

Alternative prior art systems for deriving a multimedia signature from content and for authenticating content are illustrated in Figs. 7A and 7B, respectively. The illustrated systems uses the well known public key infrastructure (PKI) in which the party providing the data has a private key 708 and a corresponding public key 714. Such a private key 708 and its corresponding public key 714 together are commonly

referred to as a "PKI key pair." Fig. 7A illustrates a system for using the private key 708 to derive a multimedia signature 710 from the original content 604. In the illustrated system, characteristic feature data (a/k/a "features") 728 are extracted by block 702 from the original content 604. Such feature data 728 can be, for example, a brightness histogram or an edge map of the original content 604. The features 728 are hashed (block 704) to derive a digest 720. The digest 720 is "signed" or encrypted by block 706 using the private key 708 to derive the multimedia signature 710.

Content purportedly signed by the system illustrated in Fig. 7A is verified using the system illustrated in Fig. 7B. In the illustrated verification system, features 730 are extracted by block 702 from the content 712 which is being verified. The features 730 are hashed by block 704 to derive a digest 722. A signature 718 — which is purportedly identical to the signature 710 derived in the signing system illustrated in Fig. 7A — is decrypted by block 724 using the public key 714 to derive decrypted data 726. If the content 712 and signature 718 are genuine, the decrypted data 726 should match the digest 722 derived from the features 730 extracted from the content 712. The digest 722 and the decrypted data 726 are therefore compared by block 716 to derive a verification result 732 which indicates successful or unsuccessful verification.

However, like the systems illustrated in Figs. 6A and 6B, the signature extraction and verification systems illustrated in Figs. 7A and 7B, respectively, can also fail to detect certain types of fraudulent manipulation. For example, it is relatively easy for an unauthorized person to fraudulently add or delete objects in an image without changing features such as the histogram or edge map of the image. The systems illustrated in Figs. 7A and 7B would therefore be unable to detect such fraudulent manipulation because the systems rely upon extraction of such features. In addition, the systems illustrated in Figs. 7A and 7B have the disadvantage of requiring a substantial amount of additional storage space for the signature 710 and 718.

Additional prior art systems for watermarking and for authenticating data are illustrated in Figs. 8A and 8B, respectively. Fig. 8A illustrates a watermarking system in which features 806 — e.g., histogram or edge map data — are extracted by block 804 from the original content 802. The features 806 are optionally randomized by block 808 to derive randomized feature data 810 which are

used as a seed input 814 of a conventional quasi-random code generator 812. Such quasi-random code generators are well-known in the art and, therefore, need not be further described. The quasi-random code generator 812 illustrated in Fig. 8A uses the randomized feature data 810 and a secret key 816 to generate watermark data 818. An embedding operation carried out by block 820 uses the secret key 816 to embed the watermark data 818 into the original content 802, thereby deriving watermarked content 822.

Referring to Fig. 8B, content 824 purportedly derived by the system illustrated in Fig. 8A is authenticated by the system illustrated in Fig. 8B. In the illustrated authentication system, watermark data 828 are extracted by block 826 from the watermarked content 824 using the same secret key 816 as was used in the watermarking system illustrated in Fig. 8A. If a randomization step 808 was used in the original watermarking procedure, the watermark data 828 is decoded/unrandomized by block 830 in the authentication system to derive unrandomized data 832. An inverse watermark generator 834 uses the secret key 816 to recover a seed 836 corresponding to the unrandomized data 832. In other words, the inverse watermark generator 834 treats the unrandomized data 832 as a watermark, and determines the seed 836 which would have been used to generate the watermark 832. Features 844 are extracted by block 838 from the watermarked content 824 and compared to the recovered seed data 836 using a correlation procedure carried out by block 840. The correlation procedure carried out by block 840 produces an authentication result 842 having a value which can range from zero to one. The result 842 indicates how much confidence should be placed in the authenticity of the watermarked content 824; a value of one indicates perfect confidence, and a value of zero indicates no confidence.

However, the watermarking and authentication systems illustrated in Figs. 8A and 8B are incompatible with PKI, and therefore require the same secret key 816 to be used in both the watermarking system and the authentication system. The secret key 816 must be distributed to any party who will be authenticating data, and this requirement increases the risk that the secrecy of the key 816 will be compromised.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a data watermarking and authentication systems which detect fraudulent manipulations of data, while permitting acceptable types of data manipulation.

It is a further object of the present invention to provide a data watermarking and authentication systems which do not require a substantial amount of additional data storage.

It is yet another object of the present invention to provide data watermarking and authentication systems which are compatible with PKI.

It is a still further object of the present invention to provide data watermarking and authentication systems which enable a user to select, with a substantial amount of precision, a level of authentication strength (i.e., reliability in detecting fraudulent manipulations) with which data will be watermarked and authenticated.

These and other objects are accomplished by the following aspects of the present invention.

In a watermarking procedure in accordance with the present invention, one or more feature codes are derived from a first set of data that is to be watermarked. One or more parity checks bits are derived from the feature codes and are included in one or more codewords. The codewords are processed by a hash operation to derive a hash result. A key code is used to sign the hash result to derive a signature. The signature is used to watermark the first set of data either by embedding the signature in the first set of data or data derived therefrom, or by storing the signature in the header of the first set of data or data derived therefrom.

In accordance with an additional aspect of the present invention, a bit plane fractionalized data set can be authenticated by deriving feature codes from respective bit plane subblocks of the data set and then comparing the feature codes to corresponding message codes derived from watermark codes that have been extracted from the aforementioned bit plane subblocks. The bit plane subblocks used to derive the feature codes are defined by respective truncation points. The data set has a first data set size and the bit plane subblocks have a second data set size. The ratio of the

first and second data set sizes is computed and used to indicate an authentication strength associated with the authentication result.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects, features, and advantages of the present invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which:

Fig. 1 is a functional block diagram illustrating an exemplary system and procedure for transforming and encoding data in accordance with the present invention;

Fig. 2A is a functional block diagram illustrating an exemplary system and procedure for watermarking data in accordance with the present invention;

Fig. 2B is a functional block diagram illustrating an additional exemplary system and procedure for watermarking data in accordance with the present invention;

Fig. 3 is a functional block diagram illustrating an exemplary system and procedure for authenticating data in accordance with the present invention;

Fig. 4 is a diagram illustrating an exemplary pattern for sequencing data in accordance with the present invention;

Fig. 5 is a functional block diagram illustrating an exemplary system and procedure for decoding and transforming data in accordance with the present invention;

Fig. 6A is a functional block diagram illustrating an exemplary prior art system and procedure for watermarking data;

Fig. 6B is a functional block diagram illustrating an exemplary prior art system and procedure for verifying data;

Fig. 7A is a functional block diagram illustrating an exemplary prior art system and procedure for signing data;

Fig. 7B is functional block diagram illustrating an exemplary prior art system and procedure for verifying data;

Fig. 8A is a functional block diagram illustrating an exemplary prior art system and procedure for watermarking data;

Fig. 8B is a functional block diagram illustrating an exemplary prior art system and procedure for authenticating data;

Fig. 9 is a diagram illustrating an exemplary computer system for implementing the systems and performing the procedures illustrated in Figs. 1-5, 6A, 6B, 7A, 7B, 8A, 8B, 11, 13, 15, and 16;

Fig. 10 is a block diagram illustrating an exemplary processing section for use in the computer system illustrated in Fig. 9;

Fig. 11 is a diagram illustrating the processing of data by an exemplary bit plane fractionalization procedure in accordance with the present invention;

Fig. 12A is a diagram illustrating bit plane fractionalized data in accordance with the present invention;

Fig. 12B is a diagram illustrating exemplary wavelet coefficients divided into subbands and codeblocks in accordance with the present invention;

Fig. 12C is a diagram illustrating an exemplary codeblock of bit plane fractionalized data divided into bit plane fractionalized subblocks in accordance with the present invention;

Fig. 13 is a functional block diagram illustrating an exemplary system and procedure for watermarking data in accordance with the present invention;

Fig. 14 is a graph illustrating an exemplary relationship between distortion and authentication bit rate for a watermarking system and procedure and an authentication system and procedure in accordance with the present invention;

Fig. 15 is a table illustrating an exemplary scheme for encoding data in accordance with the present invention; and

Fig. 16 is a functional block diagram illustrating a system and procedure for deriving a feature code in accordance with the present invention.

Throughout the figures, unless otherwise stated, the same reference numerals and characters are used to denote like features, elements, components, or portions of the illustrated embodiments.

DETAILED DESCRIPTION OF THE INVENTION

Exemplary watermarking and authentication systems in accordance with the present invention employ two stages of watermarking. In the first stage of watermarking, a set of quasi-random watermarks is generated based upon feature codes derived from various subsets of the data being watermarked. The term "feature code" is used herein to refer to a set of data that provides information regarding one or more portions of an image. For example, a feature code can be a data set representing an edge map of an image, a data set representing a brightness histogram of an image, or a sequence of bits from a rectangular region of a single bit plane of encoded, transform-domain image data. In the watermarking and authentication procedures of the present invention, each feature code is preferably an ordered sequence of bits from a square region of a single bit plane of a set of bit plane fractionalized wavelet transform coefficients, as is discussed in further detail below.

Each watermark is embedded either in the data subset from which the watermark was derived or in a different data subset. During authentication of the data, the embedded, quasi-random watermarks are used for determining which subsets of the watermarked data, if any, have been improperly manipulated.

In the second stage of watermarking, an encrypted watermark is derived from the set of feature codes as a whole, rather than on a subset-by-subset basis. The encryption is performed using a private key which is kept secret. During authentication of the data, the second stage watermark is used for determining whether or not fraudulent manipulations have been performed upon any of the watermarked data, but typically is not used for identifying which specific portion(s) or subset(s) of the data has/have been manipulated.

Data to be watermarked in accordance with the present invention is first processed by the spatial transformation and encoding system and procedure illustrated in Fig. 1. The block diagram illustrates the processing of image data, but the watermarking and authentication methods of the present invention can be applied to a variety of different types of data including, but not limited to, image data, video data, audio data, and/or other multimedia data. The procedure illustrated in Fig. 1 starts with image-domain data 102 which are subjected to a spatial domain transform by block 104 — in this example, a wavelet transform — to derive transform-domain

data 106 — in this example, wavelet transform coefficients. The transform-domain data 106 are quantized (block 108) to derive quantized transform-domain data 110 — e.g., quantized wavelet transform coefficients. In order to increase the number of quantization levels with which the content is represented, the quantized transform-domain data 110 are preferably encoded by a bit plane fractionalization procedure 112. As is discussed in further detail below, quantized transform-domain data having a larger number of quantization levels is advantageous because it provides the user with increased flexibility — i.e., a greater number of choices — in selecting the strength of authentication to be used. In the procedure illustrated in Fig. 1, the bit plane fractionalization procedure 112 is a well known encoding procedure commonly referred to as Embedded Block Coding with Optimized Truncation (EBCOT). EBCOT is, in fact, the compression engine used in the well known JPEG2000 image encoding standard, which is described in *Information Technology--JPEG2000 Image Coding System*, ISO/IEC International Standard 15444-1 (Dec. 2000). In the system and procedure illustrated in Fig. 1, the EBCOT encoding by block 112 produces a set of EBCOT codes 118.

As will be readily understood by those skilled in the art, EBCOT codes provide for convenient compression of multimedia content such as image, video, and/or audio data for transmission and storage. Data which are close to the original data can be recovered from the EBCOT codes using the system and procedure illustrated in Fig. 5. Referring to Fig. 5, EBCOT codes 502 — which, in this case, were originally derived from an image — are decoded using a conventional, inverse EBCOT (i.e., EBCOT decoding) procedure carried out by block 504 to derive a set of quantized wavelet transform coefficients 506. The quantized wavelet transform coefficients 506 can be processed using an inverse wavelet transform performed by block 508 to recover image-domain data which is close to the original image-domain data 510.

The benefits of the EBCOT encoding performed by block 112 in Fig. 1 can be further understood with reference to Fig. 11. For each transform-domain datum (e.g., wavelet coefficient 1106, illustrated in Fig. 11 as represented by three bits), EBCOT encoding derives a code 1110 whose value depends upon the value of that particular datum 1106 and the values of the surrounding data 1108 (illustrated in

Fig. 11 as represented by three bits each). The derivation of a bit plane fractionalized datum 1110 from a wavelet coefficient 1106 and its neighbors 1108 is well known in the art, and is, in fact, part of the aforementioned JPEG2000 encoding standard. The derived code 1110 is represented by more bits (12 bits in the example illustrated in Fig. 11) than any one of the wavelet coefficients 1106 and 1108 used to derive the code 1110. As a result, a block 1112 of wavelet coefficients having a particular number of bit planes 1102 (3 bit planes in the example illustrated in Fig. 11) is converted into a block 1114 of data having a larger number of bit planes 1104 (12 bit planes in the example illustrated in Fig. 11). The availability of a larger number of bit planes is beneficial for data compression and authentication, because it allows more closely spaced choices of how much compression is to be used in encoding a data set (in the case of data compression) and/or how much data is to be used for watermarking and authentication.

For example, consider the rate-distortion (R-D) function illustrated in Fig. 14. In image and/or audio compression applications, such functions are commonly used to represent the amount of distortion D as a function of "bit rate" — i.e., the ratio of the amount of data retained in the compressed data set to the amount of data in the original, uncompressed data set. If the compression is performed by truncating the data set at particular bit planes of the EBCOT coded data, then the number of choices 1402 of bit rate — and the corresponding number of possible values of distortion D — is relatively limited. However, if the bit planes are fractionalized, additional choices of truncation points, and therefore, additional bit rate choices 1404, become available. This advantage applies to authentication as well as compression. For example, consider a watermarking system and an authentication system in which the strength of the authentication is determined by the authentication bit rate — i.e., the percentage of bit planes (after truncation) used to derive features for the watermarking procedure — and in which the amount of distortion D permitted by the authentication procedure is a function of the authentication bit rate. Data having a small number of bit planes provides a small number of choices 1402 of authentication bit rate and corresponding distortion D. However, if the bit planes are fractionalized, additional choices 1404 are available.

Fig. 12B illustrates an example of an array of wavelet transform coefficients derived from an image to be watermarked in accordance with the present invention. Such wavelet data may be divided into subbands — e.g., HH1, HL1, LH1, HH2, HL2, LH2, HH3, HL3, LH3, and LL3 — representing various spatial frequency components of the image. In accordance with the present invention, the higher frequency subbands HH1, HL1, and LH1 are preferably each divided into four or more codeblocks. For example, in the wavelet coefficient set illustrated in Fig. 12B, subband HH1 is divided into four codeblocks $HH1_1$, $HH1_2$, $HH1_3$, and $HH1_4$; subband HL1 is divided into four codeblocks $HL1_1$, $HL1_2$, $HL1_3$, and $HL1_4$; and subband LH1 is divided into four codeblocks $LH1_1$, $LH1_2$, $LH1_3$, and $LH1_4$. If, for example, each of the subbands HH1, HL1, and LH1 is a square array of 256 x 256 coefficients, and is divided into square codeblocks of 64 x 64 coefficients, then each of the subbands HH1, HL1, and LH1 will consist of four codeblocks if the codeblocks do not overlap. However, if each of the subbands is a square that is larger than 256 x 256 coefficients, and is divided into square codeblocks of 64 x 64 coefficients, then each of subbands HH1, HL1, and LH1 will include more than four codeblocks.

In any case, each codeblock is encoded using the EBCOT coding, to derive a corresponding EBCOT encoded codeblock of bit plane fractionalized data. For example, codeblocks $HH1_1$, $HH1_2$, $HH1_3$, $HH1_4$, $HL1_1$, $HL1_2$, $HL1_3$, $HL1_4$, $LH1_1$, $LH1_2$, $LH1_3$, and $LH1_4$ are EBCOT encoded to derive the bit plane fractionalized subblocks 1202 illustrated in Fig. 12A. Each bit plane fractionalized codeblock 1202 is preferably a 64 x 64 square array of 4096 encoded data words. Similarly, subbands HH2, HL2, LH2, HH3, HL3, LH3, and LL3 can also be encoded to derive corresponding encoded codeblocks 1204 illustrated in Fig. 12A.

As is illustrated in Fig. 12C, each bit plane fractionalized codeblock 1202 or 1204 is preferably divided into 16 bit plane fractionalized subblocks 1206, each subblock 1206 being a 16 x 16 square array of 256 encoded data words 1208. Each encoded data word 1208 is bit plane fractionalized and therefore has a larger number of bits than any one of the wavelet transform coefficients from which it was derived.

In accordance with the present invention, wavelet transform coefficients can be watermarked using the system and procedure illustrated in Fig.

2A. In the illustrated system and procedure, EBCOT codes 118 are derived from a set of quantized wavelet transform coefficients 110 (block 112), as is discussed above with reference to Figs. 1, 12A, 12B, and 12C. Selected bit planes of selected subblocks of the EBCOT codes 118 are used to derive feature codes 204 by block 202. For example, bit plane subblock $BPSB_1$ is used to derive feature code F_1, bit plane subblock $BPSB_2$ is used to derive feature code F_2, etc. The derivation of a feature code can be further understood with reference to Fig. 4 which illustrates an exemplary 16 x 16 subblock 402 of the bits 404 in a single bit plane of a set of EBCOT codes. The illustrated bits 404 are numbered 1-256 and arranged in a left-to-right, row-by-row order. A feature code can be derived by forming a sequence of some or all of the bits 404 in the illustrated order or in any other convenient order. Alternatively, a feature code can be formed by a system and procedure 202 such as is illustrated in Fig. 16. In the illustrated system and procedure, a set 1602 of the first 239 bits of a bit plane subblock 402 (shown in Fig. 4) is formed. A set 1604 of the remaining 17 bits is combined with a set 1606 of 222 bits by padding the 17-bit data set 1604 on the right with the 222-bit data set 1606 to form a 239-bit data set 1610 (carried out in block 1608). The bit string 1602 of the first 239 bits of the bit plane subblock 402 and the newly formed bit string 1610 — which also has 239 bits — are processed by an Exclusive OR function by block 1612 to form a 239-bit feature code 1306.

Turning back to Fig. 2A, typically not all of the available bit planes are used for deriving the feature codes 204. Rather, the percentage of bits to be used for deriving feature codes is equal to an authentication bit rate 242 which can be selected by the user. For example, referring to Figs. 12A-12C, consider a case in which the selected bit rate 242 is 0.5 (i.e., 50%), and the feature codes are to be extracted from the bit plane fractionalized EBCOT codeblocks 1202 and 1204 derived respectively from (a) all of the wavelet transform codeblocks $HH1_1$, $HH1_2$, $HH1_3$, $HH1_4$, $HL1_1$, $HL1_2$, $HL1_3$, $HL1_4$, $LH1_1$, $LH1_2$, $LH1_3$, and $LH1_4$, and (b) all of subbands HH2, HL2, LH2, HH3, HL3, LH3, and LL3. The feature codes 204 are derived only from 50% of the bits of the entire set of EBCOT codeblocks 1202 and 1204. Each extracted feature code is derived — preferably using the procedure described above with respect to Fig. 16 — from an ordered sequence of bits from a particular bit plane of a particular

subblock 1206 of a particular EBCOT-encoded codeblock 1202 or 1204, but not every bit plane of the subblock 1206 is used to generate a feature code. Preferably, the feature code derivation procedure 202 uses only the most significant bit planes of each subblock 1206, up to a truncation point 1210 which can differ among the different subblocks 1206. The truncation point 1210 for each subblock 1206 is the same as the analogous truncation point used for 50% data compression in the well known EBCOT compression algorithm of the JPEG2000 standard. As an additional example, if the selected bit rate is 0.125 (i.e., 12.5%), the feature codes 204 are derived only from 12.5% of the bits of the entire set of EBCOT code blocks 1202 and 1204. The bit planes used to generate feature codes are determined by the respective truncation points of the subblocks in the same manner that the JPEG2000 compression algorithm selects truncation points for a 12.5% bit rate. The derivation of optimal truncation points for respective subblocks based on a given bit rate is well known to those skilled in the art, as is evidenced by the ISO standards document for JPEG2000 (*id.*). In the watermarking system and procedure illustrated in Fig. 2A, the selected bit rate 242 can be used as an indicator of the maximum possible authentication strength with which one can later authenticate the watermarked data produced by the watermarking system and procedure.

In the watermarking system and procedure illustrated in Fig. 2A, the feature codes 204 preferably are not used directly to form watermarks, but are first encoded using an error correction coding (ECC) procedure performed by encoder block 206 which derives parity check bits (PCBs) 208 based on the feature codes 204. For example, feature code F_1 is used to derive a group PCB_1 of parity check bits, feature code F_2 is used to derive another group PCB_2 of parity check bits, etc. The use of PCBs enhances the stability of the watermarking and authentication procedures by enabling correction of errors which may be caused by certain legitimate transformations such as requantization, wavelet transform filtration, color filtration and/or transformation, etc. Fig. 15 illustrates a look-up table (LUT) 1502 for an exemplary Hamming (7, 4) coding scheme suitable for deriving 3-bit sets of PCBs from 4-bit input codes received by an encoder. The input codes received by such an encoder, regardless of their source, are commonly referred to as "message codes." In the case of the watermarking system and procedure illustrated in Fig. 2A, the message

codes received by encoder block 206 are the respective feature codes 204 derived from EBCOT codes 118. Referring again to Fig. 15, any 4-bit message code (e.g., a 4-bit feature code) being processed according to the illustrated LUT 1502 will match one of the message codes 1504 listed in the table 1502. A particular set of PCBs 1506 is associated with each message code 1504. Each message code 1504 and its associated PCBs 1506 can be concatenated to form a codeword 1508, as is discussed below with respect to the derivation of second stage watermarks. It is to be noted that, although the LUT 1502 illustrated in Fig. 15 is configured to encode 4-bit messages, there is no particular limit to the length of messages which can be ECC encoded. For example, an LUT 1502 based on the well known BCH (255,239,2) encoding scheme can be used to process 239-bit message codes. Such encoding schemes are well known in the art, as is evidenced by W. Wesley Peterson and E.J. Weldon, Jr., *Error-Correcting Codes* (2d. ed. 1981).

Referring again to Fig. 2A, the set of PCBs derived from each feature code is used as a seed input 210 of a quasi-random code generation block 212 which generates a watermark code from each set of PCBs. For example, PCB group PCB_1 is used to derive watermark W_1; PCB group PCB_2 is used to derive watermark W_2; etc. The quasi-random watermark generation carried out by block 212 can be any conventional quasi-random code generation procedure. The quasi-random code generation procedure 212 can, for example, be an additional stage of ECC encoding. If each of the feature codes 204 is 239 bits in length, and the first ECC procedure 206 uses the well known BCH (255, 239, 2) encoding scheme (*see id.*), then each of the resulting sets of PCBs 208 will be 16 bits in length. If so, and if the quasi-random code generation procedure 212 uses the well known BCH (31, 16, 3) encoding scheme (*see id.*), then each of the resulting watermarks 214 will be a 31-bit codeword comprising a 16-bit message code (i.e., the 16 PCBs received from the first ECC procedure 206) plus 15 PCBs derived by the watermark generation procedure according to the BCH (31, 16, 3) scheme.

In any case, regardless of how the watermarks 214 are derived, they are received by a watermark input 218 of a first stage watermark embedding block 216 which embeds each of the watermarks 214 into selected bit positions of a selected, non-fractionalized bit plane of a selected subblock of the quantized wavelet

transform coefficients 110 (shown in Fig. 1), thereby deriving a set of first stage watermarked wavelet transform coefficient bits 244. The selected, non-fractionalized subblock into which the watermark is to be embedded can, optionally, be either (a) the non-fractionalized subblock corresponding to the fractionalized subblock from which the watermark was derived, or (b) a different non-fractionalized subblock. The first stage watermark embedding block 216 also provides non-watermarked wavelet transform coefficient bits 220.

With the selected subblock of quantized wavelet transform coefficients, the non-fractionalized bit plane into which the watermark is to be embedded is selected as follows. As is discussed above, for a given bit rate, the compression algorithm of the JPEG2000 standard assigns a particular truncation point to each subblock of data. For the bit plane in which the watermark will be embedded, the first stage watermark embedding by block 216 illustrated in Fig. 2A preferably selects the closest non-fractionalized bit plane that is more significant than the JPEG2000 truncation point corresponding to the selected bit rate 242. If the truncation point exactly coincides with a particular non-fractionalized bit plane, then the watermark is embedded in either that bit plane or the next more significant bit plane.

Within the selected bit plane of the selected subblock, the respective bits of the watermark are embedded by replacing existing bits of the selected bit plane in a random or quasi-random pattern based on any suitable random or quasi-random code. The wavelet transform coefficient bits 220 that are not selected for first stage watermarking are used for second stage watermarking, as is discussed in further detail below.

The watermark embedding procedure 216 can be further understood with reference to Fig. 13. As is illustrated in this block diagram, a message 1306 is derived from the bits of a particular bit plane subblock 1312 of EBCOT codes corresponding to a single subblock of a wavelet transform coefficient codeblock by block 1302. As is discussed above with respect to Figs. 4 and 16, the message 1306 can be formed by arranging selected bits of the bit plane subblock 1312 in any convenient order, or by deriving the message 1306 according to the procedure illustrated in Fig. 16. In either case, the message 1306 is processed by an ECC

function (block 206 in Fig. 2A) — based on, for example, an LUT 1502 such as is illustrated in Fig. 15 — to derive the PCBs (1308 in Fig. 13) corresponding to the message 1306. For example, if 16 x 16 subblocks are used, and BCH (255, 239, 2) encoding is applied to a 239-bit message 1306 derived from a subblock 1312 of a fractionalized bit plane using the procedure illustrated in Fig. 16, 16 PCBs are derived from the message 1306. Similarly, if BCH (31, 16, 3) encoding is applied to a 16-bit message 1306, 15 PCBs are derived from the message 1306.

As is discussed above, each set of PCBs is received by a seed input 210 in the quasi-random watermark generation block 212 illustrated in Fig. 2A. In the exemplary procedure illustrated in Fig. 13, the watermark derived from each set of PCBs is embedded by block 1310 into a non-fractionalized wavelet transform coefficient subblock 1320 that corresponds to — i.e., has been fractionalized to derive — a fractionalized bit plane subblock 1314 other than the fractionalized bit plane subblock 1312 from which the PCBs 1308 were derived. As is discussed above with respect to the embedding block 216 illustrated in Fig. 2A, the embedding by block 1310 can, for example, be performed by replacing some or all of the bits of subblock 1320 with the bits of the derived watermark. Similarly, in the illustrated example, the PCBs derived from subblock 1316 are embedded into a different non-fractionalized subblock 1322 which corresponds, e.g., to fractionalized subblock 1318 — i.e., subblock 1318 has been derived by fractionalization of subblock 1322. As is discussed above, the bit plane from which each watermark is derived is preferably immediately above (i.e., more significant than) the JPEG2000 truncation point for the subblock from which the watermark is derived, whereas the bit plane in which the watermark is embedded is preferably at the JPEG2000 truncation point for the subblock in which the watermark is embedded.

Furthermore, although Fig. 13 illustrates the embedding of a watermark into a non-fractionalized subblock not used to derive the fractionalized subblock from which the watermark was derived, a watermark can also be embedded into the non-fractionalized subblock used to derive the fractionalized subblock from which the watermark was derived — in this case, the watermark is preferably embedded in a bit plane which is more significant than the bit plane from which the watermark was derived. As is discussed above with respect to Figs. 12A and 12B, the

number of bit planes from which features are extracted is determined by the selected authentication bit rate 242 (shown in Fig. 2).

Referring again to the procedure illustrated in Fig. 2A, in addition to the first stage of watermarking — in which the first stage watermarked wavelet transform coefficient bits 244 are derived — a second stage of watermarking is performed. In the second stage portion of the watermarking procedure, each of the feature codes 204 is concatenated with its corresponding PCBs (which are within the set of PCBs 208 derived by the ECC encoding in block 206) to derive a codeword by block 222. For example, feature code F_1 is concatenated with PCB group PCB_1 to form codeword $F_1 + PCB_1$; feature code F_2 is concatenated with PCB group PCB_2 to form codeword $F_2 + PCB_2$; etc. The entire set of codewords 224 is hashed by block 226 to derive a digest 228. For the hashing procedure carried out by block 226, the codewords 224 can be arranged in any convenient order, provided that the same order is later used in the hash procedure performed by block 322 of the authentication procedure — discussed below with reference to Fig. 3. In addition, the same hash algorithm should be used in both the watermarking and authentication procedures. Examples of suitable hash algorithms include the "MD5" algorithm (which produces a 128-bit digest) and the "SHA-1" algorithm (which produces a 160-bit digest). Such hash algorithms are well known to those skilled in the art, as is evidenced by Bruce Schneier, *Applied Cryptography* (1996).

Returning now to the watermarking procedure illustrated in Fig. 2A, regardless of which type of hash algorithm is used in the hashing procedure performed by block 226, the digest 228 is "signed" by block 230 using a private key 232 which is preferably kept secret from all other parties, including the eventual recipients of the watermarked data. An example of a suitable signing procedure is the Elliptical Curve Coding procedure, which is well known to those skilled in the art, as is evidenced by William Stallings, *Cryptography and Network Security, Principles and Practice* (2d. ed. 1995). The result of the signing by block 230 is a global signature 234, which is ECC-encoded by block 246 to derive an ECC-encoded watermark 248. The ECC-encoded watermark 248 is typically 320 bits long. A second stage watermark embedding block 238 receives the watermark 248 through a watermark input 236 and embeds the watermark 248 in the non-watermarked wavelet

transform coefficient bits 220 to derive second stage watermarked wavelet transform coefficient bits 240. The ECC-encoded watermark 248 is embedded in the non-watermarked wavelet transform coefficient bits 220 based on a random pattern, typically in the same bit plane as that used to embed the first stage watermarks 214 in the first stage watermarked wavelet transform coefficient bits 244. The embedding procedure 238 embeds the second stage watermark 248 into the non-watermarked wavelet transform coefficient bits 220 by replacing selected bits of the non-watermarked wavelet coefficient bits 220 with respective bits of the ECC-encoded watermark 248.

It is well-known to those skilled in the art that alteration of certain wavelet transform coefficients tend to generate undesirable, easily visible artifacts. These particular coefficients are preferably "protected" from alteration according to human vision system (HVS) modeling techniques, and are considered "off-limits" to the watermark embedding by block 238. HVS modeling techniques are well known to those skilled in the art, as is evidenced by Anil K. Jain, *Fundamentals of Digital Image Processing* (1989). The second stage watermarked embedding by block 238 preferably does not embed any bits of the second stage watermark 248 in wavelet transform coefficients that are determined to be "protected" according to the HVS model. In any case, the watermarked wavelet transform coefficients 250 — which includes the first stage watermarked wavelet transform coefficient bits 244 and the second stage watermarked wavelet transform coefficient bits 240 — are EBCOT encoded by block 252 to derive a set of EBCOT codes 254 that have now been first stage watermarked and second stage watermarked. The first and second stage watermarked EBCOT codes 254 are thus protected from fraudulent manipulations because such manipulations can be detected by the authentication procedure illustrated in Fig. 3, as is discussed below.

Fig. 2B illustrates a variation of the watermarking system and procedure illustrated in Fig. 2A. The two procedures are identical up to the step of using the ECC codewords 248 as a watermark. However, in the system and procedure illustrated in Fig. 2B, the ECC-encoded watermark 248 is received by the watermark input 236 of a second stage watermark storage block 258, and is stored in the header of the set of watermarked and non-watermarked wavelet transform coefficients 256,

set 256 including the first stage watermarked wavelet transform coefficient bits 244 and the non-watermarked wavelet transform coefficient bits 220. The resulting first and second stage watermarked wavelet transform coefficients 240 are EBCOT encoded by block 252 to derive a set of first and second stage watermarked EBCOT codes 254. Similar to the first and second stage watermarked EBCOT codes 254 derived by the system and procedure of Fig. 2A, the first and second stage watermarked EBCOT codes 254 derived by the system and procedure of Fig. 2B are protected from unauthorized manipulations because such manipulations can be detected by the authentication system and procedure illustrated in Fig. 3.

The authentication system and procedure illustrated in Fig. 3 receives watermarked EBCOT codes 302 which purportedly — i.e., if no fraudulent manipulation has occurred — are equal to or derived from first and second stage watermarked EBCOT codes 254 generated by one of the watermarking systems and procedures illustrated in Figs. 2A and 2B. For example, the EBCOT codes 302 in Fig. 3 include the following bit plane subblocks: bit plane subblock $BPSB_1'$ which is purportedly equal to or derived from bit plane subblock $BPSB_1$; bit plane subblock $BPSB_2'$ which is purportedly equal to or derived from bit plane subblock $BPSB_2$; and other bit plane subblocks. In the authentication procedure illustrated in Fig. 3, feature codes 306 are newly derived by block 304 from the watermarked EBCOT codes 302 in the same manner as is discussed above with respect to Figs. 2A, 2B, 4, 12C, and 13. For example, newly derived feature code F_1' (derived from bit plane subblock $BPSB_1'$) illustrated in Fig. 3 is purportedly equal to feature code F_1 illustrated in Figs. 2A and 2B; and newly derived feature code F_2' (derived from bit plane subblock $BPSB_2'$) illustrated in Fig. 3 is purportedly equal to feature code F_2 illustrated in Figs. 2A and 2B. The feature codes 306 are preferably not derived from all of the fractionalized bit planes of the watermarked EBCOT codes 302, but only from bit planes more significant than respective subblock truncation points. Similarly to the watermarking systems and procedures illustrated in Figs. 2A and 2B —which define respective truncation points based on a selected watermarking bit rate 242 — the authentication system and procedure illustrated in Fig. 3 defines its respective truncation points based on a selected authentication bit rate 342. For both the watermarking procedures and the authentication procedures of the present invention,

the respective truncation points are defined in the same manner that the JPEG2000 compression standard defines truncation points based on a selected compression bit rate. As is discussed above, the derivation of optimal truncation points for respective subblocks based on a given bit rate is well known to those skilled in the art, as is evidenced by the ISO standards document for JPEG2000: *Information Technology--JPEG2000 Image Coding System,* ISO/IEC International Standard 15444-1 (Dec. 2000). The bit rate 342 used in the authentication procedure of Fig. 3 can be any rate less than or equal to the selected bit rate 242 used to watermark the data 110 in the watermarking procedures illustrated in Figs. 2A and 2B. The authentication bit rate 342, which is the ratio of the number of untruncated bits used to derive the feature codes 306 to the total number of bits in the set of EBCOT codes 302, can be used to indicate the authentication strength of the illustrated authentication procedure.

The watermarked EBCOT codes 302 are decoded by an inverse EBCOT operation carried out by block 354 to derive a set of wavelet transform coefficients 356. A set of first stage embedded watermarks 310 is extracted by block 308 from the wavelet transform coefficients 356 using the same bit rate 342 as is used to derive the feature codes 306. As is discussed above with respect to the watermarking systems and procedures illustrated in Figs. 2A and 2B, first stage watermarks 214 are typically embedded in the quantized wavelet transform coefficients 110 (shown in Figs. 1, 2A, and 2B) by replacing selected bits of selected, non-fractionalized bit planes of selected subblocks of the coefficients 110 with respective bits of the first stage watermark 214 (shown in Figs. 2A and 2B). Therefore, first stage embedded watermarks 310 can be extracted by collecting the same replacement bits and arranging the bits in their original order to form the watermarks 310. If the watermarked EBCOT codes 302 are authentic and have not been improperly altered, the first stage watermarks 310 extracted from the wavelet transform coefficients 356 should be identical to the first stage watermarks 214 (shown in Figs. 2A and 2B) that were embedded in the quantized wavelet transform coefficients 110 during the watermarking procedure — preferably one of the procedures illustrated in Figs. 2A and 2B. Accordingly, each of the extracted watermarks 310 is associated with a particular one of the newly derived feature codes 306, because if no fraudulent manipulations have occurred, each of the extracted

watermarks 310 is derived from and equal to a watermark — i.e., one of the watermarks 214 in the watermarking system and procedure illustrated in Fig. 2A or Fig. 2B — that was originally derived from a feature code (one of the feature codes 204 in Figs. 2A and 2B) which, as is discussed above, should be equal to a particular newly derived feature code (one of the feature codes 306 illustrated in Fig. 3). For example, watermark W_1' is associated with feature code F_1' because watermark W_1' (in Fig. 3) was purportedly derived from watermark W_1 (in Figs. 2A and 2B) which was derived from feature code F_1 (in Figs. 2A and 2B) which should be equal to feature code F_1', as is discussed above. Similarly, watermark W_2' is associated with feature code F_2' because watermark W_2' (in Fig. 3) was purportedly derived from watermark W_2 (in Figs. 2A and 2B) which was derived from feature code F_2 (in Figs. 2A and 2B) which should be equal to feature code F_2', as is discussed above.

Therefore, because each of the extracted first stage watermarks 310 (in Fig. 3) should be identical to a corresponding one of the embedded first stage watermarks 214 (in Figs. 2A and 2B), it should be possible, by reversing the first stage watermark generation process, to recover an ECC message code that either (a) equals the original feature code used to derive the embedded first stage watermark, or (b) has corresponding PCBs that are identical to the PCBs corresponding to the original feature code. For simplicity, and because the authentication system need not make any assumption regarding whether the extracted code is the original feature code or a message code having the same PCBs as the original feature code, the recovered code will be referred to herein as simply a "message code." As is discussed in further detail below, the recovered message code can then be compared to the newly derived feature code that is associated with the extracted watermark, to determine whether there is a successful match. A match between the recovered message code and the corresponding newly derived feature code is considered to be successful if the recovered message code and the newly derived feature code have the same corresponding PCBs in the chosen ECC encoding scheme.

To recover the respective message codes, the first stage watermark generation process is reversed as follows. Each of the extracted watermarks 310 is processed by an inverse quasi-random code generation procedure carried out by block 338 having a seed output 346, to derive PCBs 340. If no unauthorized manipulations

have occurred, the newly derived PCBs 340 should be identical to the PCBs 208 (shown in Figs. 2A and 2B) that were used to generate the watermarks 214 in the quasi-random watermark generation block 212 of the watermarking system and procedure of Figs. 2A and 2B. For example, PCB group PCB_1' (in Fig. 3) should be identical to PCB group PCB_1 which was derived from bit plane subblock $BPSB_1$ (in Figs. 2A and 2B); and PCB group PCB_2' (in Fig. 3) should be identical to PCB group PCB_2 which was derived from bit plane subblock $BPSB_2$ (in Figs. 2A and 2B).

Each group of PCBs within the resulting set of PCBs 340 is then decoded by an error correction decoding (i.e., inverse ECC) procedure 348 to generate a message code. Each message code in the resulting set of message codes 350 has thus been derived from a group of PCBs which, as is discussed above, should be identical to the PCBs derived from a particular bit plane subblock of the EBCOT codes 118 (in Figs. 2A and 2B) that were subjected to the watermarking procedure. For example, message code MC_1 (in Fig. 3) is derived from PCB group PCB_1' (in Fig. 3) which should be identical to the PCB group PCB_1 (in Figs. 2A and 2B) that was derived from bit plane subblock $BPSB_1$ (in Figs. 2A and 2B). Similarly, message code MC_2 (in Fig. 3) is derived from the PCB group PCB_2' (in Fig. 3) which should be identical to the PCB group PCB_2 (in Figs. 2A and 2B) that was derived from bit plane subblock $BPSB_2$ (in Figs. 2A and 2B). Because two or more different message codes can be associated with (i.e., can be ECC encoded to derive) the same group of PCBs — as is apparent in the exemplary ECC encoding scheme illustrated in Fig. 15, discussed above — message codes MC_1, MC_2, etc. are not necessarily identical to the respective newly derived feature codes F_1', F_2', etc. However, as is discussed above, each message code should be either equal to its corresponding newly derived feature code, or matched to the same group of PCBs in the chosen ECC encoding scheme. For example, referring to the exemplary scheme illustrated in Fig. 15, if newly derived feature code F_1' equals 0110 —which corresponds to (i.e., is associated with) a PCB group having a value of 011 — then if no improper manipulations have occurred, MC_1 should equal either 0110 or 1000, because each of these two message code values corresponds to a PCB group having a value of 011.

The newly derived feature codes 306 are therefore compared on a subblock-by-subblock basis to the corresponding message codes 350 derived from the

extracted watermarks 310 (step 312). For each feature code and message code being compared, the comparison procedure 312 determines whether the PCBs associated with the message code equal the PCBs associated with the feature code. The comparison procedure 312 preferably also includes conventional ECC error correction block 352 to correct spurious errors. If any of the blocks has one or more newly derived feature codes which fail to compare properly to the corresponding message codes (block 316), the comparison procedure carried out by block 312 identifies the failed blocks 320. On the other hand, if all of the blocks pass (block 314), the comparison procedure 312 derives ECC codewords 318 based on the newly derived feature codes 306 — each codeword including a newly derived feature code and a set of PCBs derived from that feature code — and the ECC codewords 318 are hashed in block 322 to derive a digest 324. For example, a codeword $F_1' + PCB_1''$ is formed by concatenating feature code F_1' with a set PCB_1'' of PCBs derived from the feature code F_1'. Similarly, a codeword $F_2' + PCB_2''$ is formed by concatenating feature code F_2' with a set PCB_2'' of PCBs derived from the feature code F_2'.

In addition, a set of ECC codewords 358 is extracted by block 330 from the wavelet transform coefficients 356 that were derived from the EBCOT codes 302 by block 354, discussed above. If the second stage watermarks were purportedly embedded —using the second stage watermark embedding by block 238 illustrated in Fig. 2A — in the watermarked wavelet transform coefficients 250 from which the EBCOT codes 302 were derived, and block 238 embedded the second stage watermark bits by simply replacing selected bits of the wavelet transform coefficients according to a random or quasi-random pattern, as is discussed above with respect to Fig. 2A, then the second stage watermark extraction procedure 330 illustrated in Fig. 3 derives ECC codewords 358 by locating the embedded bits within the newly derived wavelet transform coefficients 356 and arranging the bits in their original order based on the aforementioned random or quasi-random pattern. On the other hand, if the second stage watermarks were purportedly stored — using the second stage watermark storage by block 258 illustrated in Fig. 2B — in the header of the wavelet transform coefficients 240 from which the EBCOT codes 302 were derived, then the second stage watermark extraction by block 330 illustrated in Fig. 3 derives ECC codewords 358 by collecting them from the header of the set of newly derived

wavelet transform coefficients 356. It is to be noted that the set of wavelet transform coefficients from which the second stage watermark is to be extracted is typically determined by the same bit rate 342 used to perform the feature code derivation by block 304 and the first stage watermark extraction by block 308. If there has been no unauthorized manipulation/modification of the data, the extracted ECC codewords 358 should be identical to the ECC encoded watermark 248 (shown in Figs. 2A and 2B) used to watermark the data in the watermarking system and procedure (preferably one of the systems and procedures illustrated in Figs. 2A and 2B).

The extracted ECC codewords 358 are decoded using an error correction decoding (i.e., inverse ECC) procedure 360 to derive a decoded second stage watermark 332. If there has been no fraudulent manipulation/modification of the data, the decoded second stage watermark 332 should be identical to the global signature 234 embedded in the data in the watermarking system and procedure illustrated in Fig. 2A or Fig. 2B. The decoded second stage watermark 332 — which is a signature extracted from the wavelet transform coefficients 356 derived from the watermarked EBCOT codes 302 — is decrypted by block 334 using a PKI public key 344 which corresponds to the private key 232 used to encrypt the digest by block 228 in the watermarking system and procedure illustrated in Fig. 2A or Fig. 2B. The decrypted signature 336 is then compared to the newly derived digest 324 on a bit-by-bit basis by block 326 to derive an authentication result 328. The authentication result 328 indicates whether the watermarked EBCOT codes 302 have been fraudulently manipulated.

It will be appreciated by those skilled in the art that the systems and procedures illustrated in Figs. 1, 2A, 2B, 3-5, 11, 12A-C, 13, 15, and 16 may be implemented on various standard computer platforms having the requisite computing power and operating under the control of suitable software, the design of which software will be apparent to such skilled artisans from the description contained herein. In some cases, at least portions of the systems and procedures described herein may be implemented using dedicated special purpose hardware.

Figs. 9 and 10 illustrate typical computer hardware suitable for performing the methods of the present invention. Referring to Fig. 9, the computer system includes a processing section 910, a display 920, a keyboard 930, and a

communications peripheral device 940 such as a modem. The system typically includes a digital pointer 990 (e.g., a "mouse") and can also include other input devices such as an optical scanner 950 for scanning an image medium 900. In addition, the system can include a printer 960. The computer system typically includes a hard disk drive 980 and one or more additional disk drives 970 which can read and write to computer readable media such as magnetic media (e.g., diskettes or removable hard disks), or optical media (e.g., CD-ROMS or DDS). The disk drives 970 and 980 are used for storing data, operation system software, and application software.

Fig. 10 is a functional block diagram which further illustrates the processing section 910. The processing section 910 generally includes a processing unit 1010, control logic 1020, and a memory unit 1030. Preferably, the processing section 910 also includes a timer 1050 and input/output ports 1040. The processing section 910 can also include a co-processor 1060, depending on the microprocessor used in the processing unit. Control logic 1020 provides, in conjunction with processing unit 1010, the control necessary to handle communications between memory unit 1030 and input/output ports 1040. Timer 1050 provides a timing reference signal for processing unit 1010 and control logic 1020. Co-processor 1060 provides an enhanced ability to perform complex computations in real time, such as those required by cryptographic algorithms.

Memory unit 1030 can include different types of memory, such as volatile and non-volatile memory and read-only and programmable memory. For example, as shown in Fig. 10, memory unit 1030 can include read-only memory (ROM) 1031, electrically erasable programmable read-only memory (EEPROM) 1032, and random-access memory (RAM) 1033. Different computer processors, memory configurations, data structures and the like can be used to practice the present invention, and the invention is not limited to a specific platform. For example, although the processing section 910 is illustrated in Figs. 9 and 10 as part of a computer system, the processing section 910 and/or its components can be incorporated into an imager such as a digital video camera or a digital still-image camera.

Software defined by the description herein may be written in a wide variety of programming languages, as will be appreciated by those skilled in the art.

Although the present invention has been described in connection with specific exemplary embodiments, it should be understood that various changes, substitutions, and alterations can be made to the disclosed embodiments without departing from the spirit and scope of the invention, the scope being set forth in the appended claims.

WE CLAIM:

1. A method for watermarking data, comprising the steps of:

deriving at least one feature code from a first set of data;

deriving at least one parity check bit from the at least one feature code;

including the at least one parity check bit in at least one codeword;

processing the at least one codeword by a first hash operation for deriving a first hash result; and

using a first key code to sign the first hash result for deriving signature data.

2. A method according to claim 1, wherein the at least one codeword further includes the at least one feature code.

3. A method according to claim 1, further comprising:

processing a second set of data by a domain transformation procedure for deriving a set of transform-domain data;

quantizing the set of transform-domain data for deriving a set of quantized data having a first number of quantization levels; and

encoding the set of quantized data by a bit plane fractionalization procedure for deriving the first set of data, the first set of data having a second number of quantization levels, the second number of quantization levels being greater than the first number of quantization levels.

4. A method according to claim 3, wherein the domain transformation procedure comprises a wavelet transform, the set of transform-domain data comprising a set of wavelet transform coefficients.

5. A method according to claim 1, further comprising the steps of:

using the at least one parity check bit as a seed of a quasi-random code generation procedure for deriving watermark data;

embedding the watermark data in selected bit positions of a truncated portion of the first set of data for deriving a first set of watermarked data;

deriving a set of non-watermarked data corresponding to the first set of watermarked data; and

deriving a second set of watermarked data by at least one of: (a) embedding data derived from the signature data in the set of non-watermarked data,

and (b) concatenating the data derived from the signature data with a data set that includes both the first set of watermarked data and the set of non-watermarked data.

6. A method according to claim 5, further comprising:

processing a second set of data by a domain transformation procedure for deriving a set of transform-domain data;

quantizing the set of transform-domain data for deriving a set of quantized data having a first number of quantization levels; and

encoding the set of quantized data by a bit plane fractionalization procedure for deriving the first set of data, the first set of data having a second number of quantization levels, the second number of quantization levels being greater than the first number of quantization levels.

7. A method according to claim 6, wherein the at least one codeword further includes the at least one feature code.

8. A method according to claim 6, wherein the domain transformation procedure comprises a wavelet transform, the set of transform-domain data comprising a set of wavelet transform coefficients.

9. A method according to claim 5, wherein the at least one codeword further includes the at least one feature code.

10. A method for authenticating a set of watermarked data derived from the second set of watermarked data of claim 5, the method comprising the steps of:

deriving a set of feature codes from the set of watermarked data;

extracting a set of watermark codes from data derived from the set of watermarked data, each of the watermark codes being associated with a respective one of the feature codes derived from the set of watermarked data;

performing a first set of authentication steps on each of the feature codes derived from the set of watermarked data and on a watermark code associated with the each of the feature codes derived from the set of watermarked data, the first set of authentication steps comprising:

using an inverse quasi-random code generation procedure to derive a respective set of at least one parity check bit from the watermark code associated with the each of the feature codes derived from the set of watermarked data,

error correction decoding the respective set of at least one parity check bit for deriving a respective message code, the respective message code having associated therewith a first corresponding set of at least one parity check bit, each of the feature codes derived from the set of watermarked data having associated therewith a second corresponding set of at least one parity check bit, and

determining whether the first corresponding set of at least one parity check bit equals the second corresponding set of at least one parity check bit; and

if the first set of authentication steps being performed on every one of the feature codes derived from the set of watermarked data and on the watermark code associated with the every one of the feature codes determines that the first corresponding set of at least one parity check bit equals the second corresponding set of at least one parity check bit, performing a second set of authentication steps comprising:

deriving a set of codewords having at least a respective one of the feature codes derived from the set of watermarked data, each codeword of the set of codewords further including a set of at least one parity check bit corresponding to the respective one of the feature codes included in the codeword,

processing the set of codewords by a second hash operation for deriving a second hash result,

extracting a second set of signature data from the data derived from the set of watermarked data,

using a second key code to decrypt the second set of signature data for deriving decrypted signature data, the second key code comprising a public key code, the first key code comprising a private key code, the public and private key codes forming a PKI key pair, and

comparing the second hash result with the decrypted signature data for deriving an authentication result.

11. A method for deriving image-domain data from the set of watermarked data of claim 10, the method comprising the steps of:

decoding the set of watermarked data by an inverse bit plane fractionalization procedure for deriving a set of quantized transform-domain data, the

set of watermarked data having a first number of quantization levels, the set of quantized transform-domain data having a second number of quantization levels, the first number of quantization levels being greater than the second number of quantization levels; and

processing the set of quantized transform-domain data by an inverse domain transformation for deriving the image-domain data.

12. A method according to claim 11, wherein the set of quantized, transform-domain data comprises a set of wavelet transform coefficients, and the inverse domain transformation procedure comprises an inverse wavelet transform.

13. A method for authenticating data, comprising the steps of:

using an authentication bit rate equal to a ratio of a first data set size and a second data set size to determine respective bit plane truncation points defining respective bit plane subblocks of a data set, the data set having the first data set size and the bit plane subblocks of the data set having the second data set size;

deriving feature codes from the respective bit plane subblocks of the data set; and

comparing the feature codes to corresponding message codes for deriving a first authentication result, the message codes being derived from watermark codes extracted from transform-domain data derived from the bit plane subblocks of the data set.

14. A method according to claim 13, further comprising, if the first authentication result indicates that the data set is authentic, the steps of:

deriving a respective codeword from each one of the feature codes, the respective codeword comprising the feature code from which the respective codeword is derived and at least one parity check bit derived from the feature code;

processing the codewords by a hash operation for deriving a hash result;

extracting error correction encoded data from the transform-domain data;

error correction decoding the error correction encoded data for deriving signature data;

using a key code to decrypt the signature data for deriving decrypted

signature data; and

comparing the hash result to the decrypted signature data for deriving a second authentication result.

15. A method for deriving image-domain data from the data set of claim 14, the method comprising the steps of:

decoding the data set by an inverse bit plane fractionalization procedure for deriving a set of quantized transform-domain data, the data set having a first number of quantization levels, the set of quantized transform-domain data having a second number of quantization levels, the first number of quantization levels being greater than the second number of quantization levels; and

processing the set of quantized transform-domain data by an inverse domain transformation for deriving the image-domain data.

16. A method according to claim 15, wherein the set of quantized transform-domain data comprises wavelet transform coefficients, and the inverse domain transformation comprises an inverse wavelet transform.

17. A method for deriving image-domain data from the data set of claim 13, the method comprising the steps of:

decoding the data set by an inverse bit plane fractionalization procedure for deriving a set of quantized transform-domain data, the data set having a first number of quantization levels, the set of quantized transform-domain data having a second number of quantization levels, the first number of quantization levels being greater than the second number of quantization levels; and

processing the set of quantized transform-domain data by an inverse domain transformation for deriving the image-domain data.

18. A method according to claim 17, wherein the set of quantized transform-domain data comprises wavelet transform coefficients, and the inverse domain transformation comprises an inverse wavelet transform.

19. An apparatus for watermarking data, comprising:

a first processor for deriving at least one feature code from a first set of data;

a second processor for deriving at least one parity check bit from the at least one feature code;

a third processor for including the at least one parity check bit in at least one codeword;

a fourth processor for hashing the at least one codeword for deriving a first hash result; and

a fifth processor for using a first key code to sign the first hash result for deriving signature data.

20. An apparatus according to claim 19, wherein the at least one codeword further includes the at least one feature code.

21. An apparatus according to claim 19, further comprising:

a sixth processor for domain transforming a second set of data for deriving a set of transform-domain data;

a seventh processor for quantizing the set of transform-domain data for deriving a set of quantized data having a first number of quantization levels; and

an eighth processor for encoding the set of quantized data by bit plane fractionalization for deriving the first set of data, the first set of data having a second number of quantization levels, the second number of quantization levels being greater than the first number of quantization levels.

22. An apparatus according to claim 21, wherein the sixth processor comprises a wavelet transform processor, the set of transformed-domain data comprising a set of wavelet transform coefficients.

23. An apparatus according to claim 19, further comprising:

a sixth processor responsive to the at least one parity check bit as a seed for quasi-random code generation to derive watermark data;

a seventh processor for embedding the watermark data in selected bit positions of a truncated portion of the first set of data for deriving a first set of watermarked data;

deriving a set of non-watermarked data corresponding to the first set of watermarked data; and

an eighth processor for deriving a second set of watermarked data by at least one of: (a) embedding data derived from the signature data in the set of non-watermarked data, and (b) concatenating the data derived from the signature data with

a data set that includes both the first set of watermarked data and the set of non-watermarked data.

24. An apparatus according to claim 23, further comprising:

a ninth processor for domain transforming a second set of data for deriving a set of transform-domain data;

a tenth processor for quantizing the set of transform-domain data for deriving a set of quantized data having a first number of quantization levels; and

an eleventh processor for encoding the set of quantized data by bit plane fractionalization for deriving the first set of data, the first set of data having a second number of quantization levels, the second number of quantization levels being greater than the first number of quantization levels.

25. An apparatus according to claim 24, wherein the at least one codeword further includes the at least one feature code.

26. An apparatus according to claim 24, wherein the ninth processor comprises a wavelet transform processor, and the set of transform-domain data comprises a set of wavelet transform coefficients.

27. An apparatus according to claim 23, wherein the at least one codeword further includes the at least one feature code.

28. An apparatus for authenticating a set of watermarked data derived from the second set of watermarked data of claim 23, the apparatus comprising:

a first processor for deriving a set of feature codes from the set of watermarked data;

a second processor for extracting a set of watermark codes from data derived from the set of watermarked data, each of the watermark codes being associated with a respective one of the feature codes derived from the set of watermarked data;

a third processor for authenticating each of the feature codes derived from the set of watermarked data, the third processor further for authenticating a watermark code associated with the each of the feature codes derived from the set of watermarked data, the third processor comprising, for each one of the feature codes derived from the set of watermarked data:

a fourth processor for inverse quasi-random code generation to

derive a respective set of at least one parity check bit from the watermark code associated with the each of the feature codes derived from the set of watermarked data,

a fifth processor for error correction decoding the respective set of at least one parity check bit for deriving a respective message code, the respective message code having associated therewith a first corresponding set of at least one parity check bit, each one of the feature codes derived from the set of watermarked data having associated therewith a second corresponding set of at least one parity check bit, and

a sixth processor for determining whether the first corresponding set of at least one parity check bit equals the second corresponding set of at least one parity check bit; and

a seventh processor responsive to the sixth processor determining that the first corresponding set of at least one parity check bit equals the second corresponding set of at least one parity bit for every one of the feature codes derived from the set of watermarked data and for the watermark code associated with the every one of the feature codes, for deriving a set of codewords by including in each one of the codewords: 1) at least a respective one of the feature codes derived from the set of watermarked data, and 2) at least the parity check bit derived from the respective one of the feature codes,

an eighth processor responsive to the seventh processor deriving a set of codewords for hashing the set of codewords for deriving a second hash result,

a ninth processor for responsive to the eighth processor deriving a hash result for extracting a second set of signature data from the data derived from the set of watermarked data,

a tenth processor responsive to the ninth processor extracting signature data for using a second key code to decrypt the second set of signature data for deriving decrypted signature data, the second key code comprising a public key code, the first key code comprising a private key code, the public and private key codes forming a PKI key pair, and

an eleventh processor responsive to the tenth processor deriving

decrypted signature data for comparing the second hash result with the decrypted signature data for deriving an authentication result.

29. An apparatus for deriving image-domain data from the set of watermarked data of claim 28, comprising:

a first processor for decoding the set of watermarked data by inverse bit plane fractionalization for deriving a set of quantized transform-domain data, the set of watermarked data having a first number of quantization levels, the set of quantized transform-domain data having a second number of quantization levels, the first number of quantization levels being greater than the second number of quantization levels; and

a second processor for inverse domain transforming the set of quantized transform-domain data for deriving the image-domain data.

30. An apparatus according to claim 29, wherein the set of quantized transform-domain data comprises a set of wavelet transform coefficients, and the second processor comprises an inverse wavelet transform processor.

31. An apparatus for authenticating data, comprising:

a first processor for using an authentication bit rate equal to a ratio of a first data set size and a second data set size to determine respective bit plane truncation points defining respective bit plane subblocks of a data set, the data set having the first data set size and the bit plane subblocks of the data set having the second data set size;

a second processor for deriving feature codes from the respective bit plane subblocks of the data set; and

a third processor for comparing the feature codes to corresponding message codes for deriving a first authentication result, the message codes being derived from the watermark codes extracted from transform-domain data derived from the bit plane subblocks of the data set.

32. An apparatus according to claim 31, further comprising:

a fourth processor for deriving, if the first authentication result indicates that the data set is authentic, a respective codeword from each one of the feature codes, the respective codeword comprising the feature code from which the respective codeword is derived and at least one parity check bit derived from the

feature code;

a fifth processor for hashing the codewords for deriving a hash result;

an sixth processor for extracting error correction encoded data from the transform-domain data;

a seventh processor for error correction decoding the error correction encoded data for deriving signature data;

an eighth processor for using a key code to decrypt the signature data for deriving decrypted signature data; and

a ninth processor for comparing the hash result to the decrypted signature data for deriving a second authentication result.

33. An apparatus for deriving image-domain data from the data set of claim 32, comprising:

a first processor for decoding the data set by inverse bit plane fractionalization for deriving a set of quantized transform-domain data, the data set having a first number of quantization levels, the set of quantized transform-domain data having a second number of quantization levels, the first number of quantization levels being greater than the second number of quantization levels; and

a second processor for inverse domain transforming the set of quantized transform-domain data for deriving the image-domain data.

34. An apparatus according to claim 33, wherein the set of quantized transform-domain data comprises wavelet transform coefficients, and the second processor comprises an inverse wavelet transform processor.

35. An apparatus for deriving image-domain data from the data set of claim 31, comprising:

a first processor for decoding the data set by inverse bit plane fractionalization for deriving a set of quantized transform-domain data, the data set having a first number of quantization levels, the set of quantized transform-domain data having a second number of quantization levels, the first number of quantization levels being greater than the second number of quantization levels; and

a second processor for inverse domain transforming the set of quantized transform-domain data for deriving the image-domain data.

36. An apparatus according to claim 35, wherein the set of quantized transform-domain data comprises wavelet transform coefficients, and the second processor comprises an inverse wavelet transform processor.

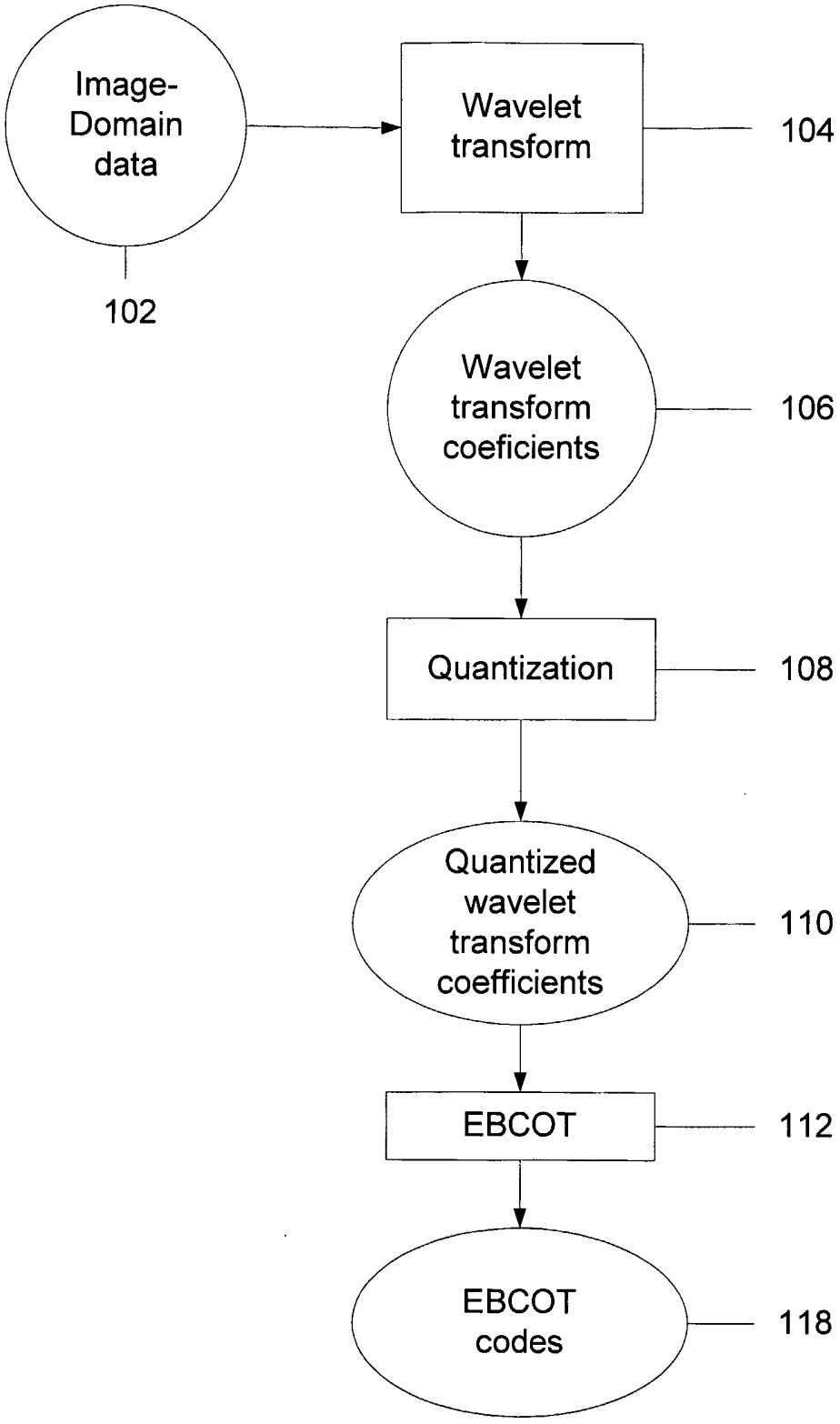


Image-Domain data
102
Wavelet transform
104
Wavelet transform coeficients
106
Quantization
108
Quantized wavelet transform coefficients
110
EBCOT
112
EBCOT codes
118

Fig. 1

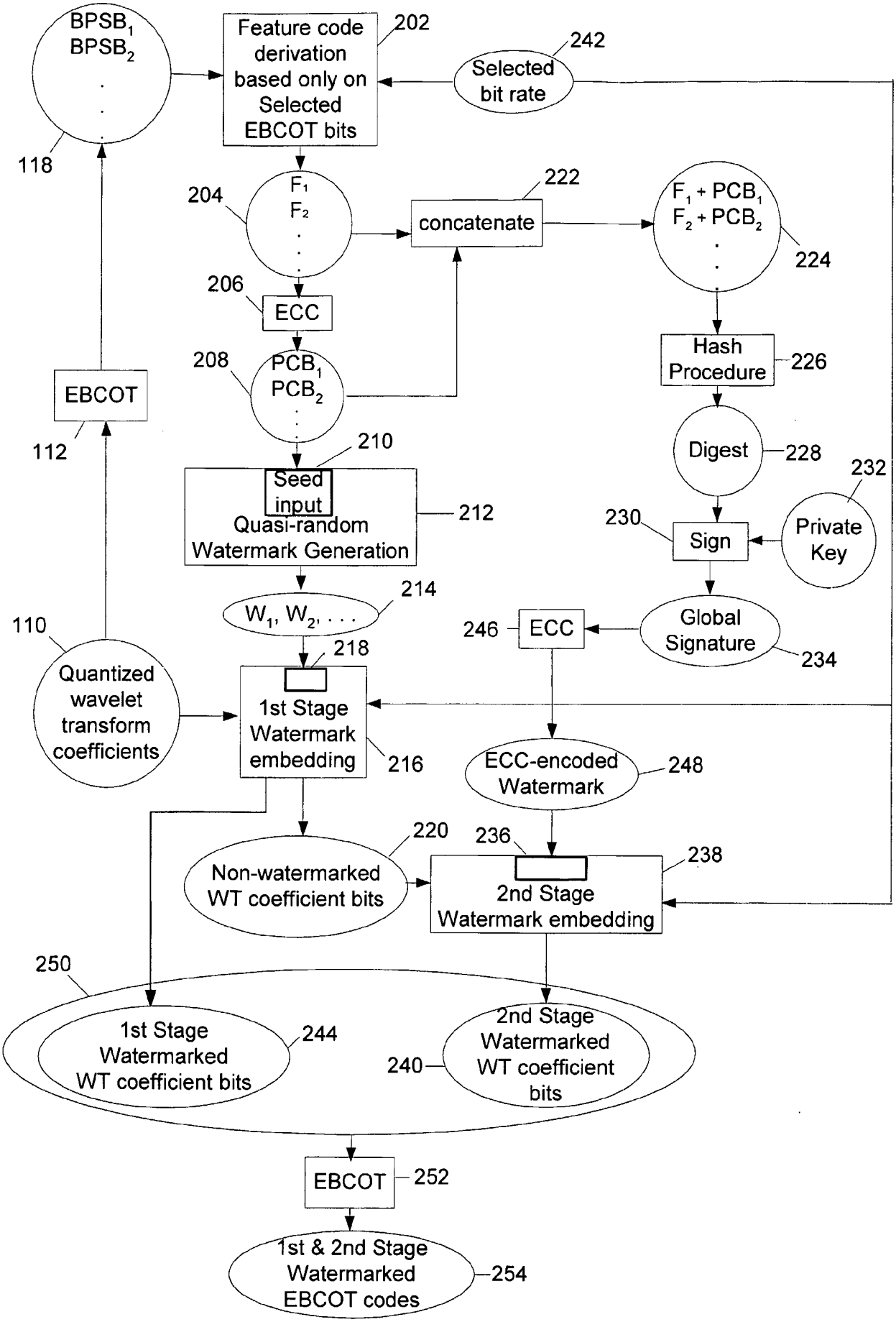

BPSB1
BPSB2
118
Feature code derivation based only on Selected EBCOT bits
202
242
Selected bit rate
204
F1
F2
222
concatenate
F1 + PCB1
F2 + PCB2
224
206
ECC
208
PCB1
PCB2
Hash Procedure
226
EBCOT
112
Digest
228
210
Seed input
Quasi-random Watermark Generation
212
232
230
Sign
Private Key
W1, W2, . . .
214
246
ECC
Global Signature
234
110
218
Quantized wavelet transform coefficients
1st Stage Watermark embedding
216
ECC-encoded Watermark
248
220
236
238
Non-watermarked WT coefficient bits
2nd Stage Watermark embedding
250
1st Stage Watermarked WT coefficient bits
244
240
2nd Stage Watermarked WT coefficient bits
EBCOT
252
1st & 2nd Stage Watermarked EBCOT codes
254

FIG. 2A

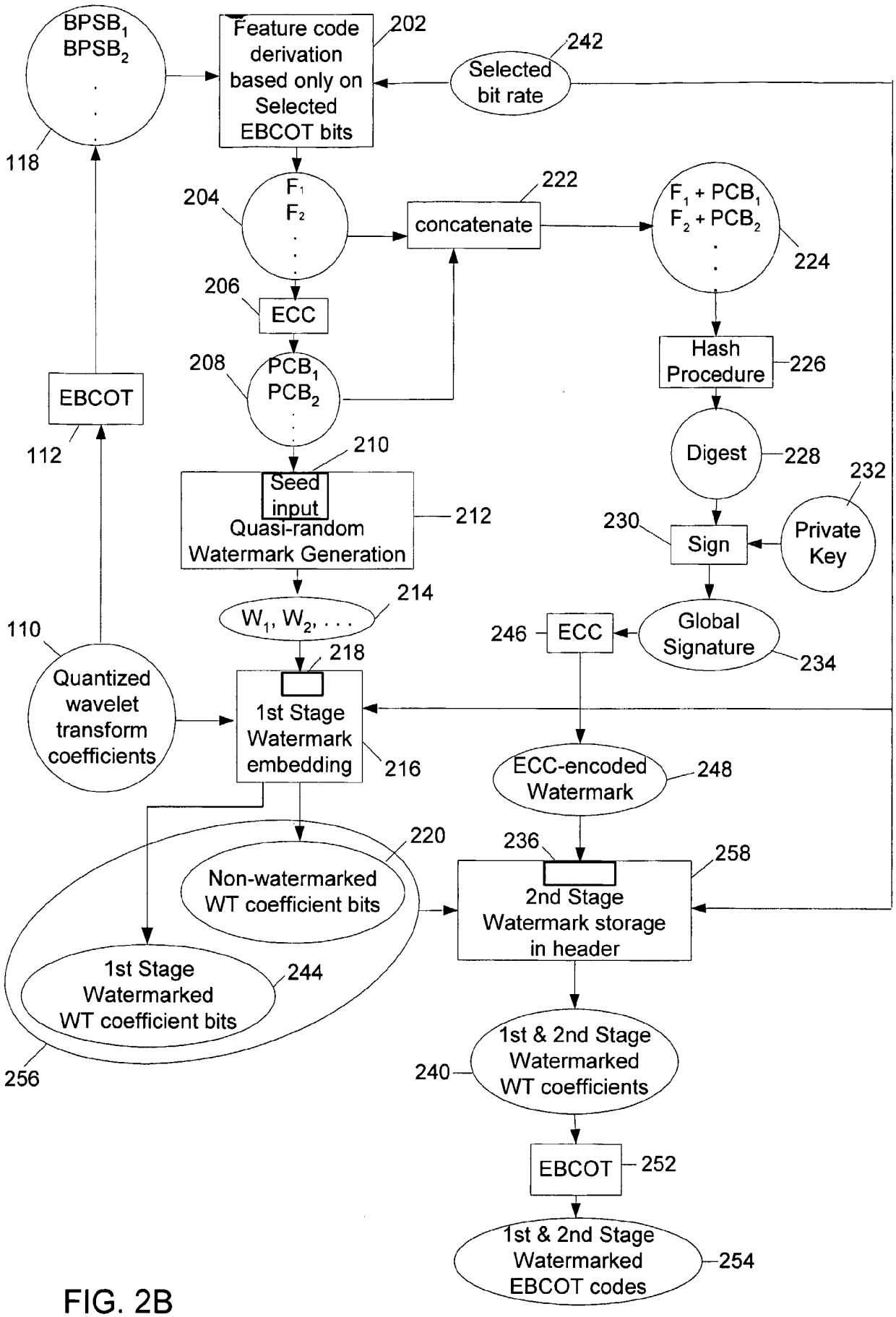

BPSB1
BPSB2
118
Feature code derivation based only on Selected EBCOT bits
202
Selected bit rate
242
204
F1
F2
concatenate
222
F1 + PCB1
F2 + PCB2
224
206
ECC
208
PCB1
PCB2
EBCOT
112
Hash Procedure
226
Digest
228
232
Private Key
230
Sign
210
Seed input
Quasi-random Watermark Generation
212
W1, W2, . . .
214
110
218
Quantized wavelet transform coefficients
1st Stage Watermark embedding
216
246
ECC
Global Signature
234
ECC-encoded Watermark
248
220
236
258
Non-watermarked WT coefficient bits
2nd Stage Watermark storage in header
1st Stage Watermarked WT coefficient bits
244
256
1st & 2nd Stage Watermarked WT coefficients
240
EBCOT
252
1st & 2nd Stage Watermarked EBCOT codes
254

FIG. 2B

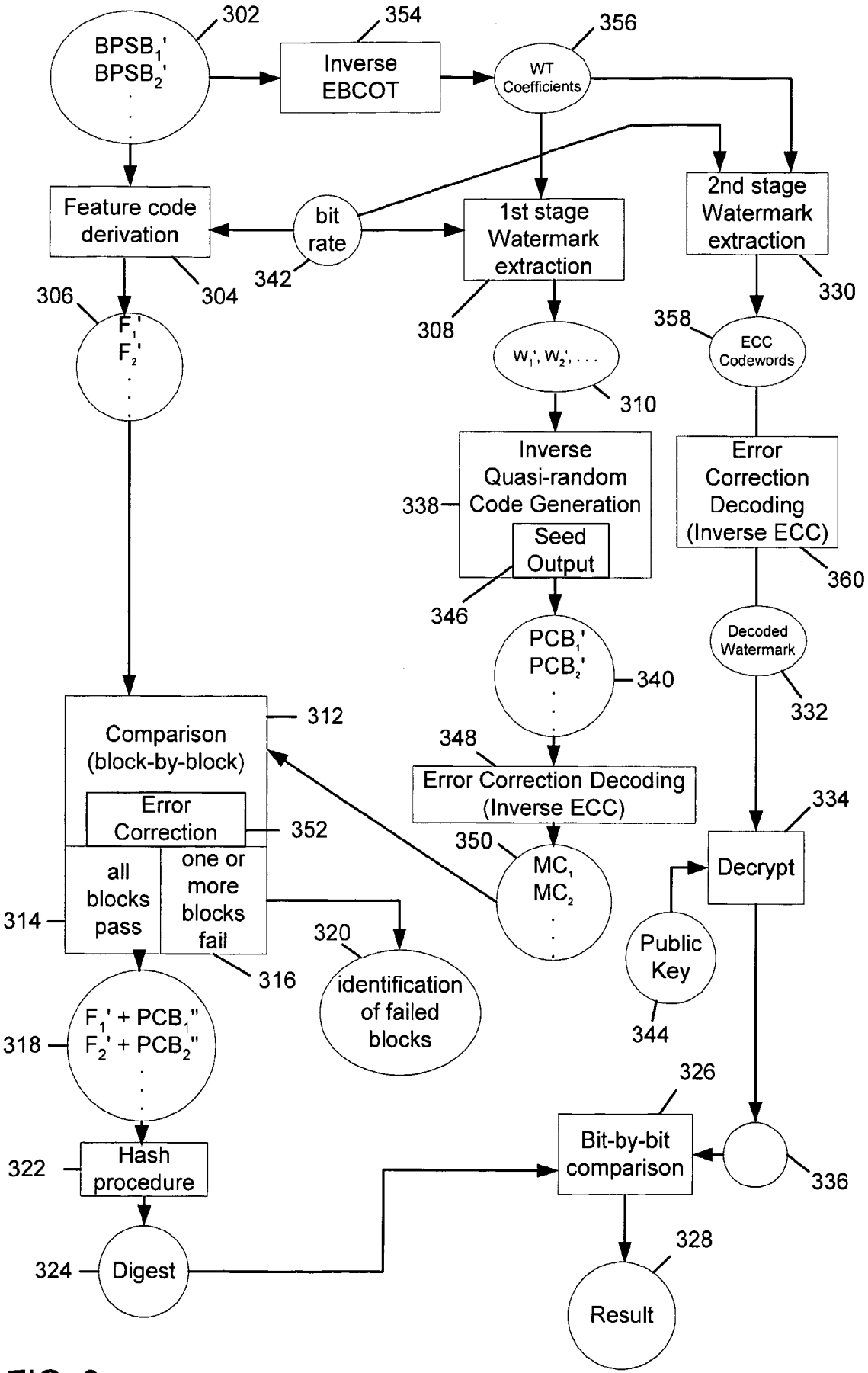

BPSB1'
BPSB2'
302
Inverse EBCOT
354
WT Coefficients
356
Feature code derivation
304
bit rate
342
1st stage Watermark extraction
308
2nd stage Watermark extraction
330
306
F1'
F2'
W1', W2', . . .
310
358
ECC Codewords
Inverse Quasi-random Code Generation
338
Seed Output
346
Error Correction Decoding (Inverse ECC)
360
PCB1'
PCB2'
340
Decoded Watermark
332
Comparison (block-by-block)
312
Error Correction
352
348
Error Correction Decoding (Inverse ECC)
334
350
MC1
MC2
Decrypt
all blocks pass
one or more blocks fail
314
316
320
identification of failed blocks
Public Key
344
F1' + PCB1"
F2' + PCB2"
318
Hash procedure
322
326
Bit-by-bit comparison
336
Digest
324
328
Result

FIG. 3

402

404

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16
#17	#18	#19	#20	#21	#22	#23	#24	#25	#26	#27	#28	#29	#30	#31	#32
#33	#34	#35	#36	#37	#38	#39	#40	#41	#42	#43	#44	#45	#46	#47	#48
#49	#50	#51	#52	#53	#54	#55	#56	#57	#58	#59	#60	#61	#62	#63	#64
#65	#66	#67	#68	#69	#70	#71	#72	#73	#74	#75	#76	#77	#78	#79	#80
#81	#82	#83	#84	#85	#86	#87	#88	#89	#90	#91	#92	#93	#94	#95	#96
#97	#98	#99	#100	#101	#102	#103	#104	#105	#106	#107	#108	#109	#110	#111	#112
#113	#114	#115	#116	#117	#118	#119	#120	#121	#122	#123	#124	#125	#126	#127	#128
#129	#130	#131	#132	#133	#134	#135	#136	#137	#138	#139	#140	#141	#142	#143	#144
#145	#146	#147	#148	#149	#150	#151	#152	#153	#154	#155	#156	#157	#158	#159	#160
#161	#162	#163	#164	#165	#166	#167	#168	#169	#170	#171	#172	#173	#174	#175	#176
#177	#178	#179	#180	#181	#182	#183	#184	#185	#186	#187	#188	#189	#190	#191	#192
#193	#194	#195	#196	#197	#198	#199	#200	#201	#202	#203	#204	#205	#206	#207	#208
#209	#210	#211	#212	#213	#214	#215	#216	#217	#218	#219	#220	#221	#222	#223	#224
#225	#226	#227	#228	#229	#230	#231	#232	#233	#234	#235	#236	#237	#238	#239	#240
#241	#242	#243	#244	#245	#246	#247	#248	#249	#250	#251	#252	#253	#254	#255	#256

Fig. 4

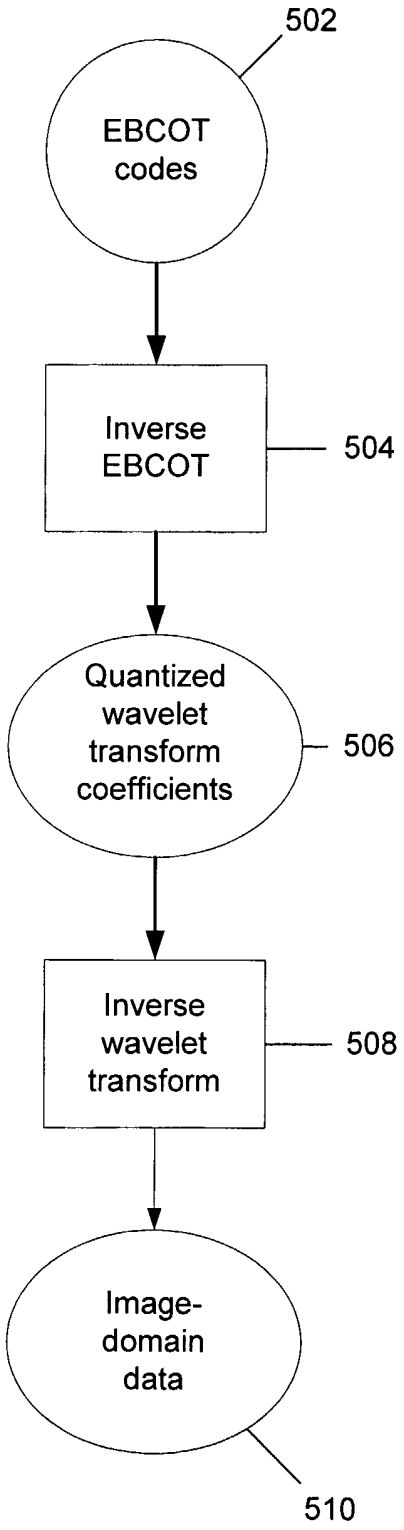

502
EBCOT codes
Inverse EBCOT
504
Quantized wavelet transform coefficients
506
Inverse wavelet transform
508
Image-domain data
510

Fig. 5

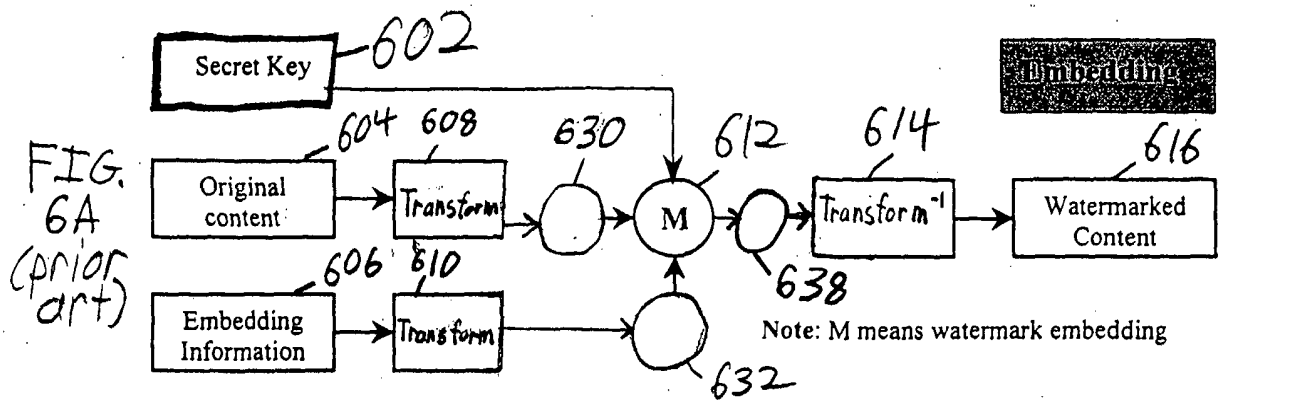


FIG. 6A (prior art)

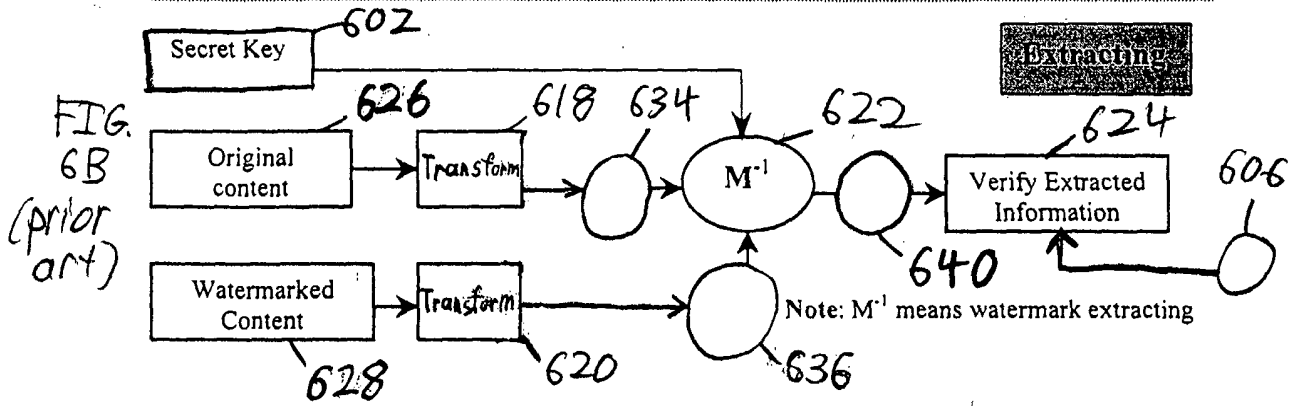


FIG. 6B (prior art)

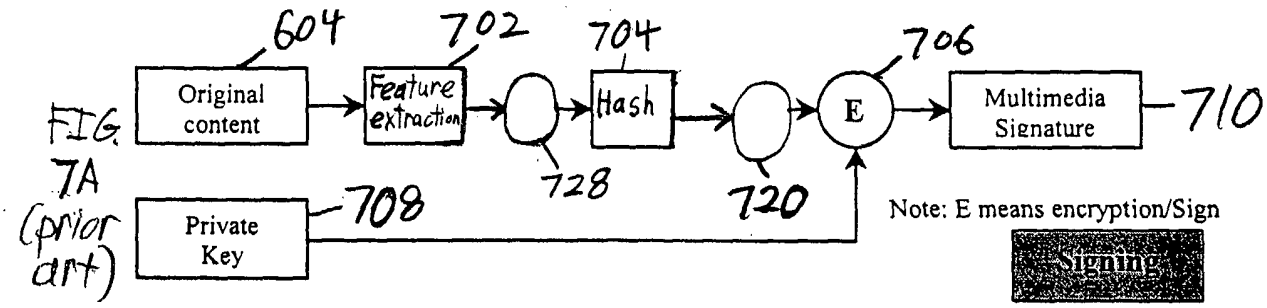


FIG. 7A (prior art)

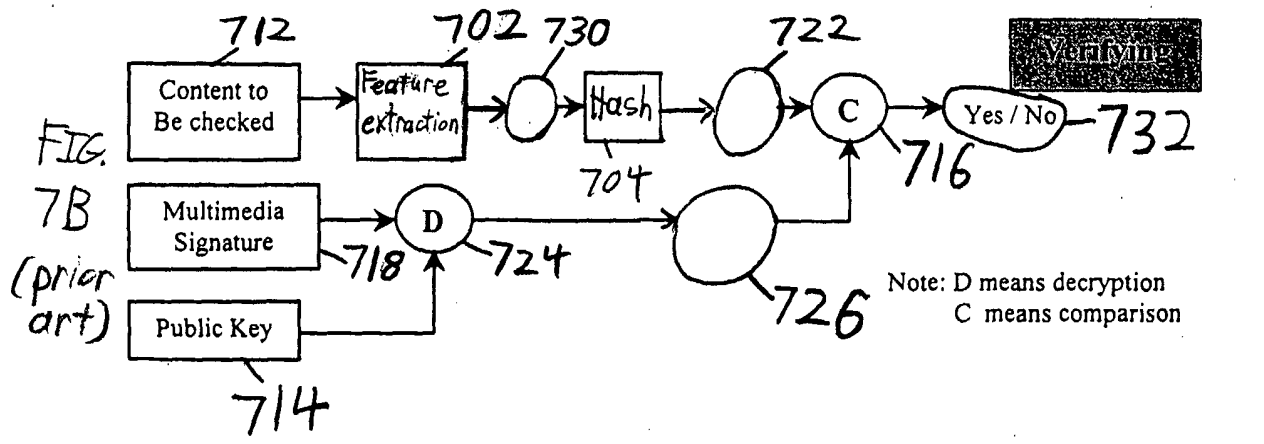


FIG. 7B (prior art)

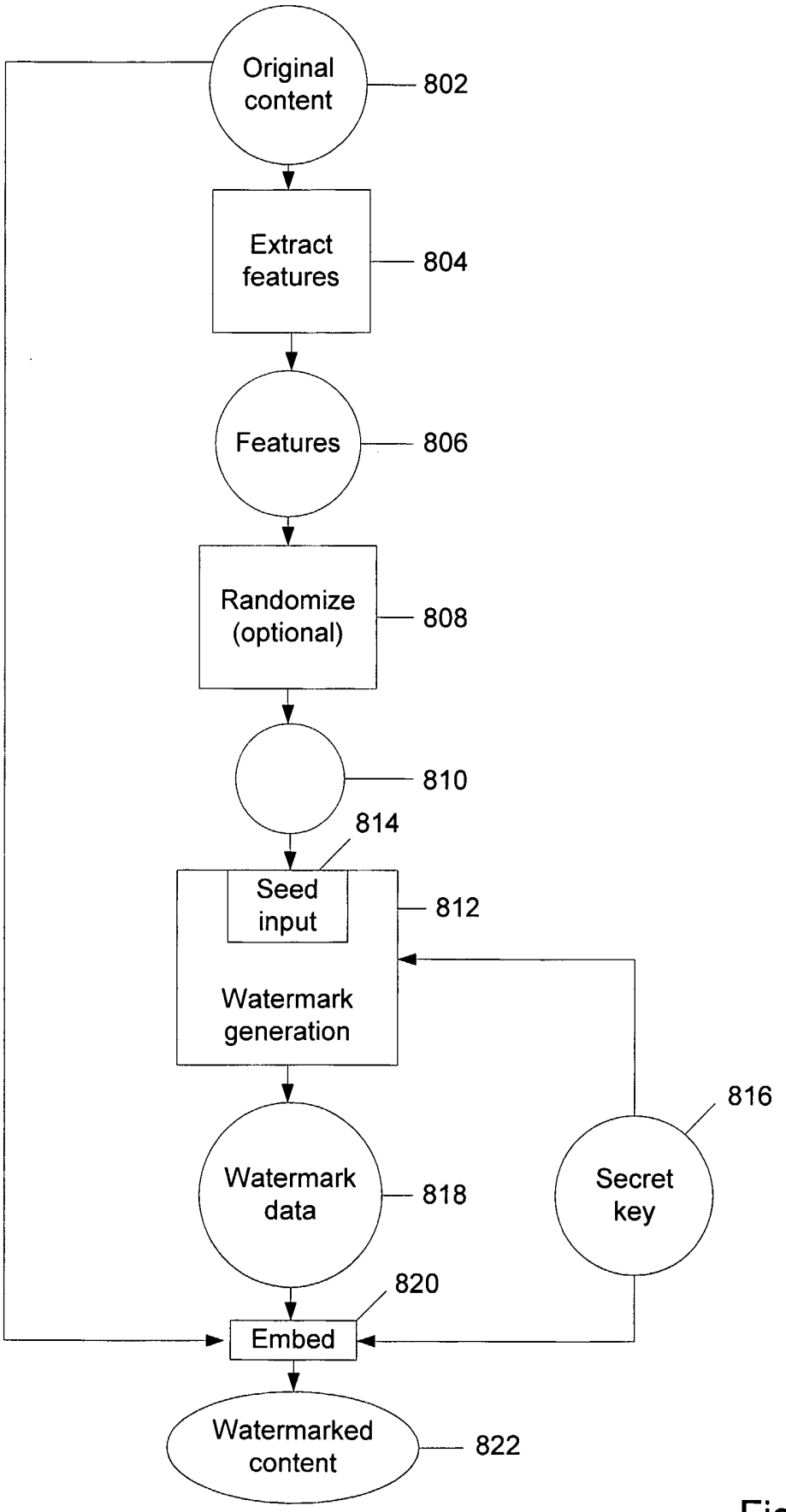


Fig. 8A
(prior art)

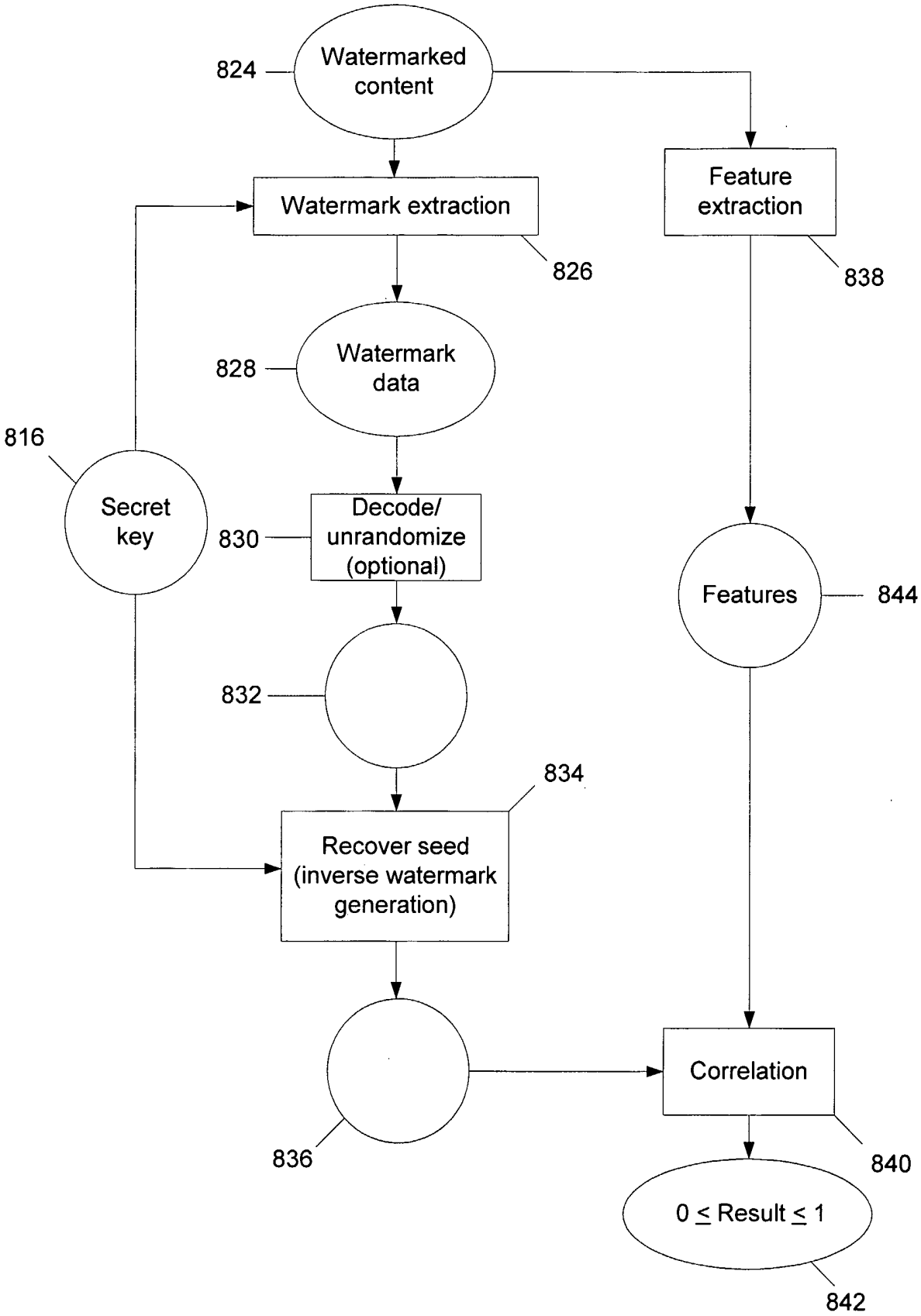


Fig. 8B
(prior art)

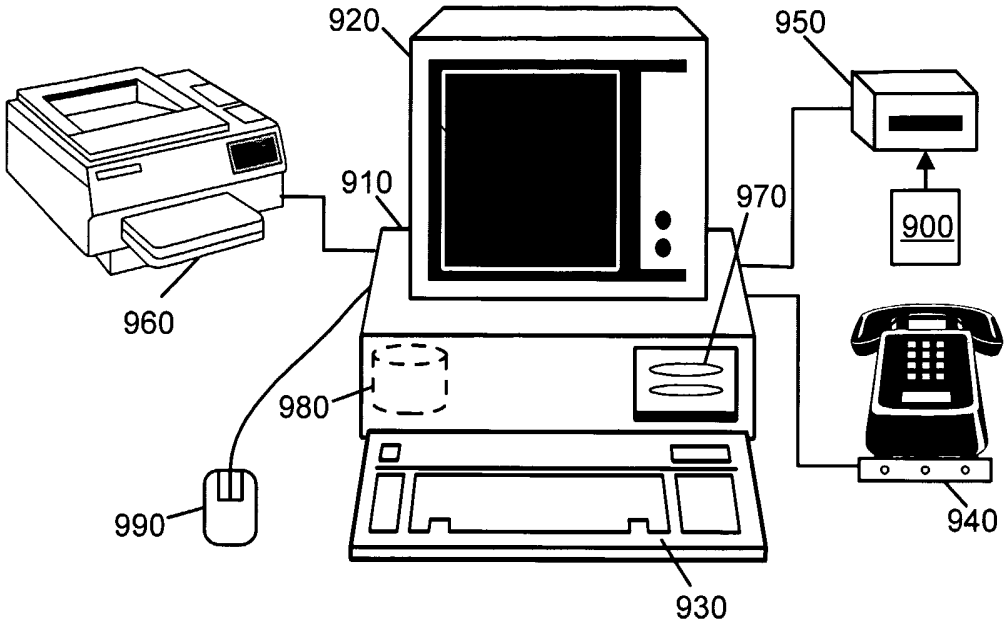

920
950
910
970
900
960
980
990
940
930

FIG. 9

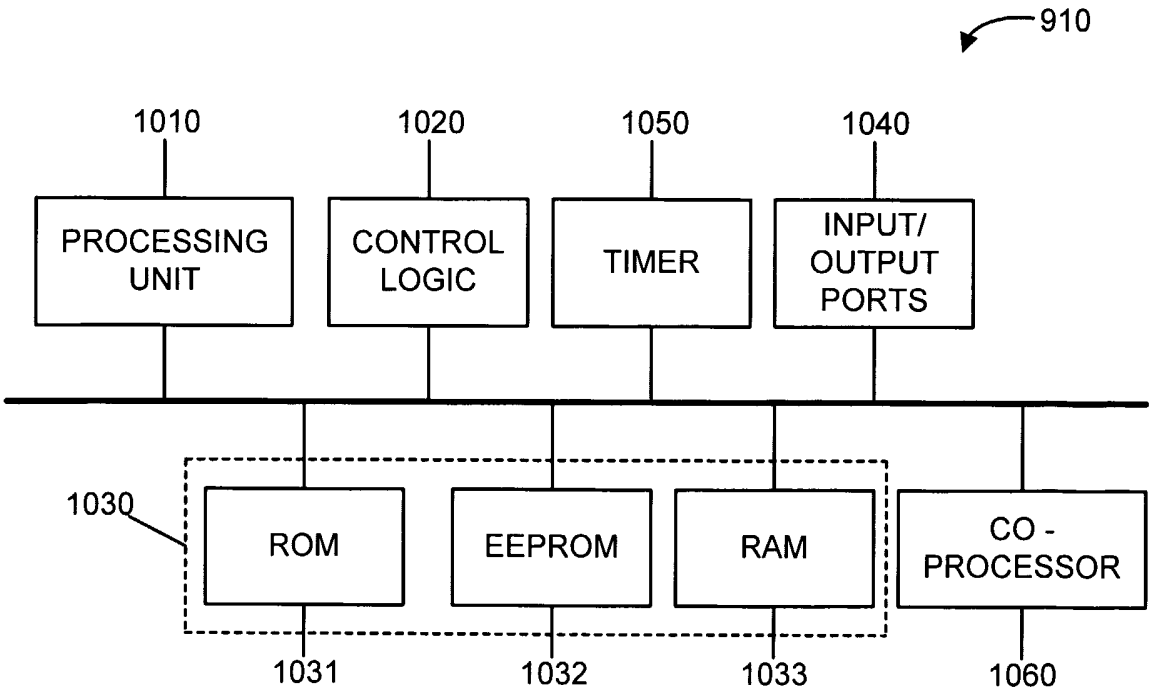

910
1010
1020
1050
1040
PROCESSING UNIT
CONTROL LOGIC
TIMER
INPUT/ OUTPUT PORTS
1030
ROM
EEPROM
RAM
CO - PROCESSOR
1031
1032
1033
1060

FIG. 10

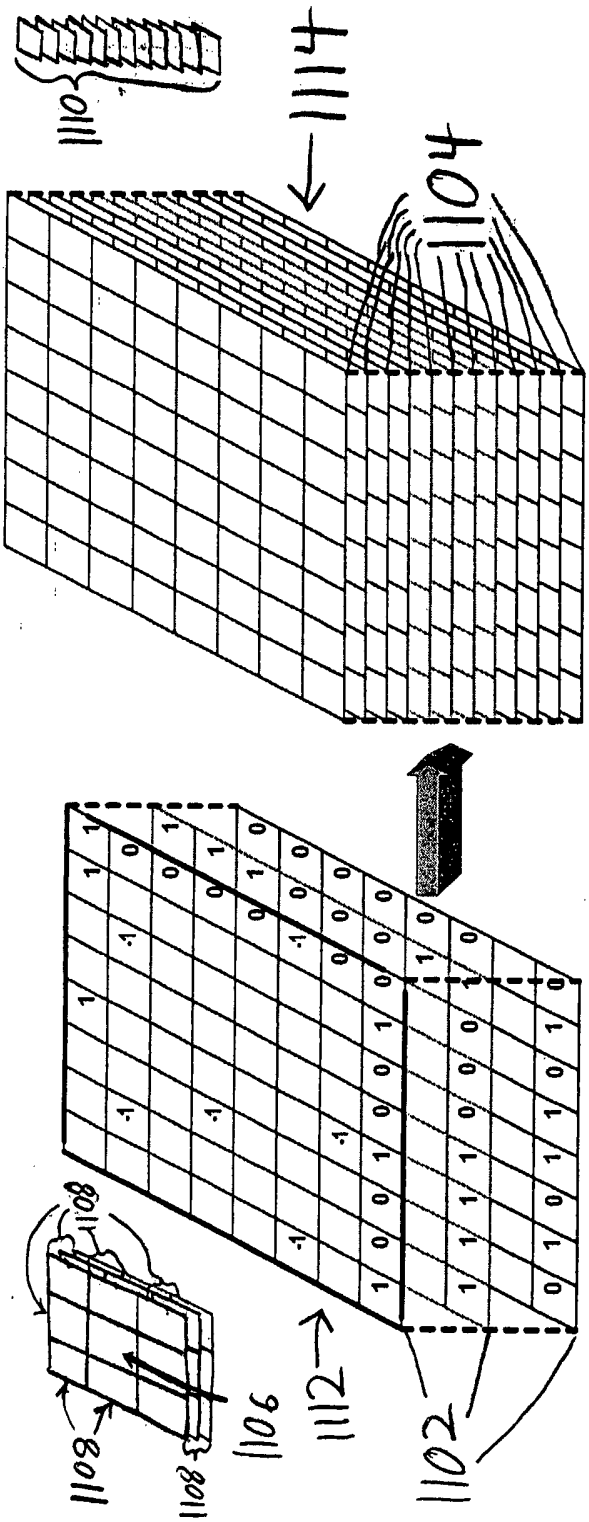
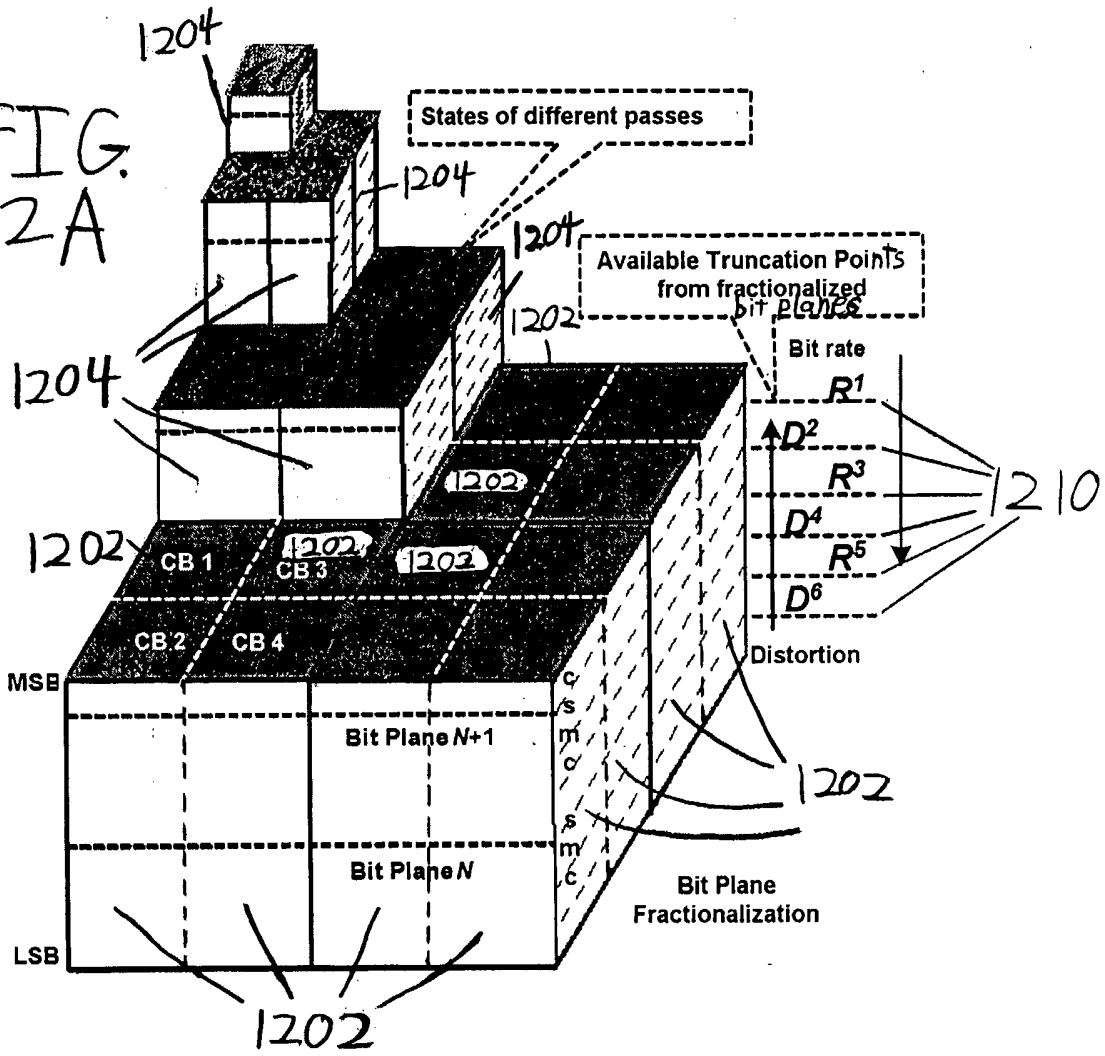


FIG. 11

FIG. 12A



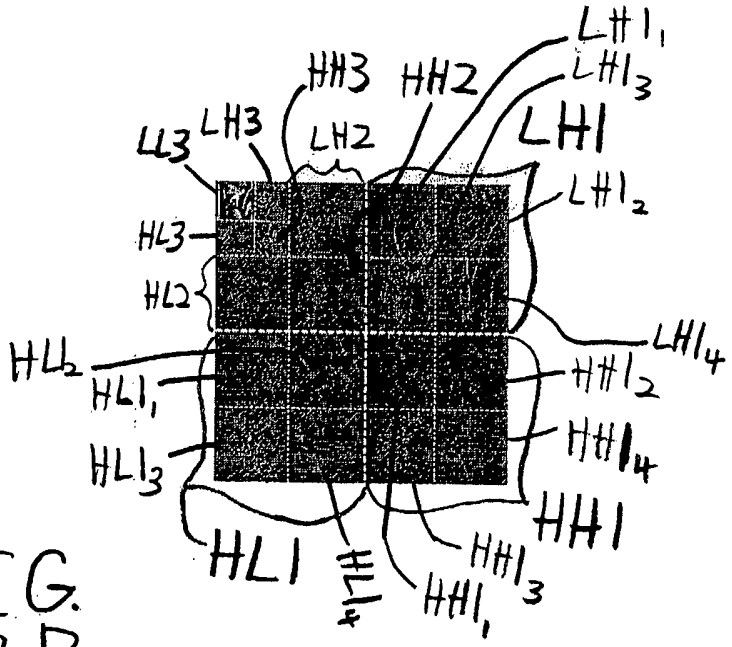


FIG. 12B

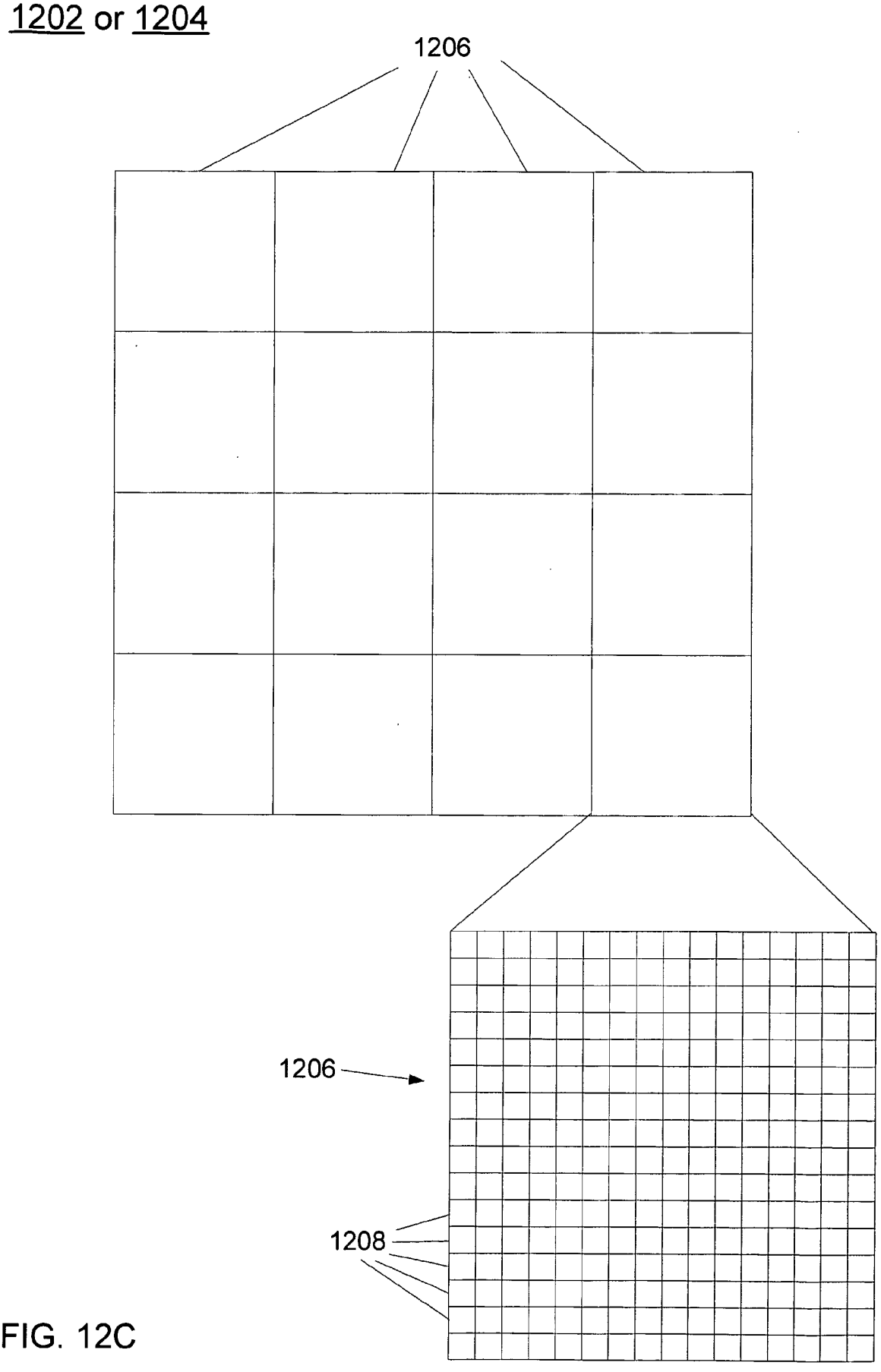


FIG. 12C

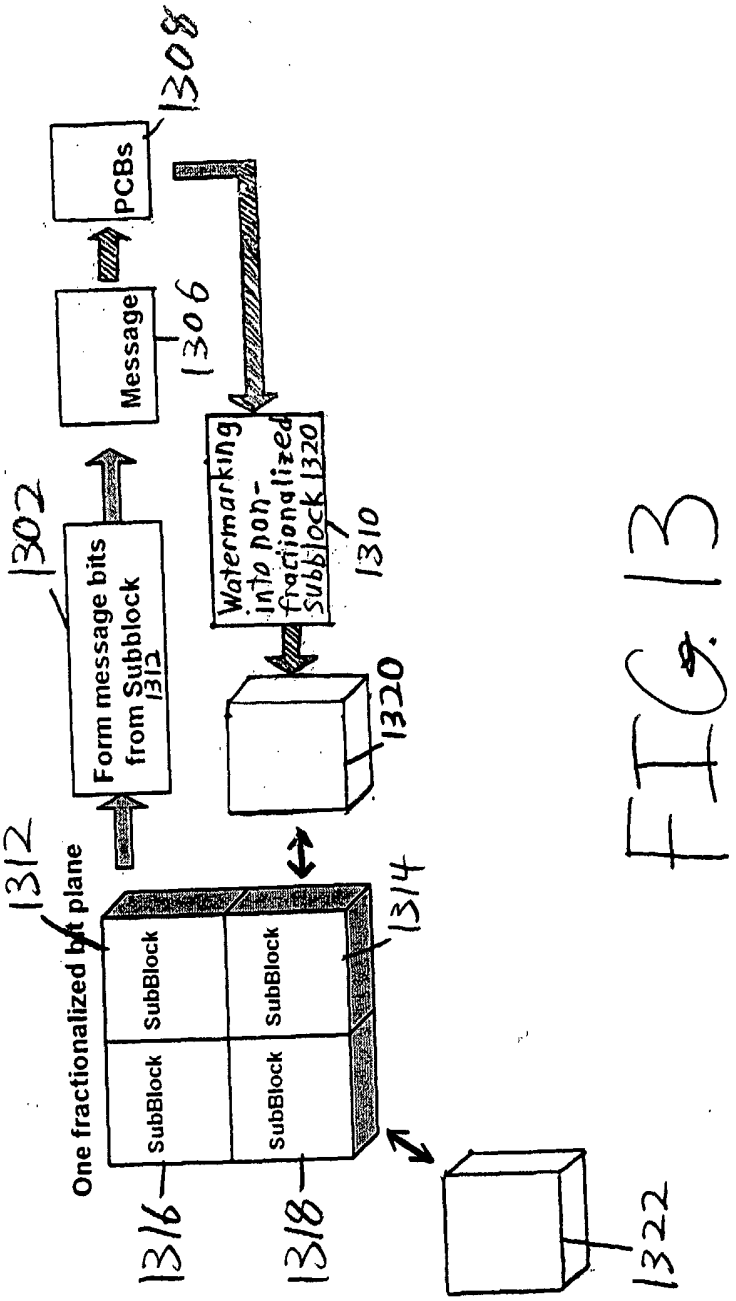

1308
PCBs
1306
Message
1302
Form message bits from Subblock 1312
1310
Watermarking into non-fractionalized Subblock 1320
1320
1312
One fractionalized bit plane
SubBlock
SubBlock
SubBlock
SubBlock
1314
1316
1318
1322

FIG. 13

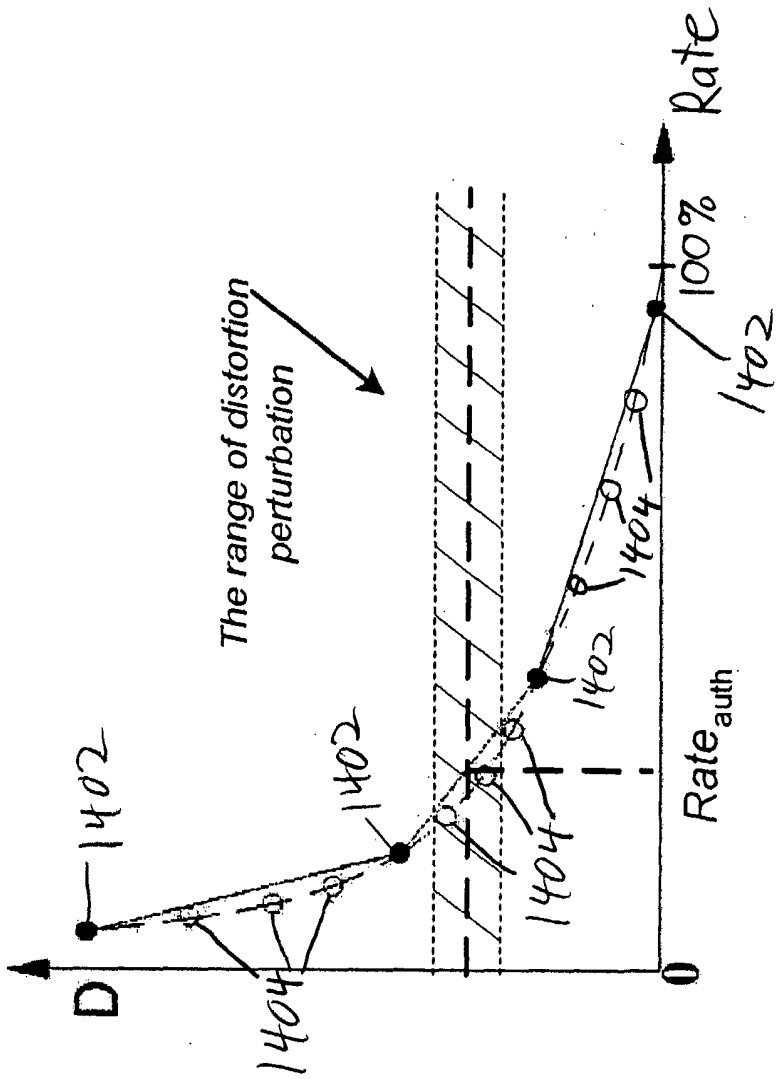

D
0
Rate
100%
Rate$_{auth}$
The range of distortion perturbation
1402
1404

FIG. 14

Hamming (7, 4) Coding

Message	PCBs	Codeword
0000	000	0000000
0001	111	0001111
0010	110	0010110
0011	001	0011001
0100	101	0100101
0101	010	0101010
0110	011	0110011
0111	100	0111100
1000	011	1000011
1001	100	1001100
1010	101	1010101
1011	010	1011010
1100	110	1100110
1101	001	1101001
1110	000	1110000
1111	111	1111111

1502 1504 1506 1508

Fig. 15

202

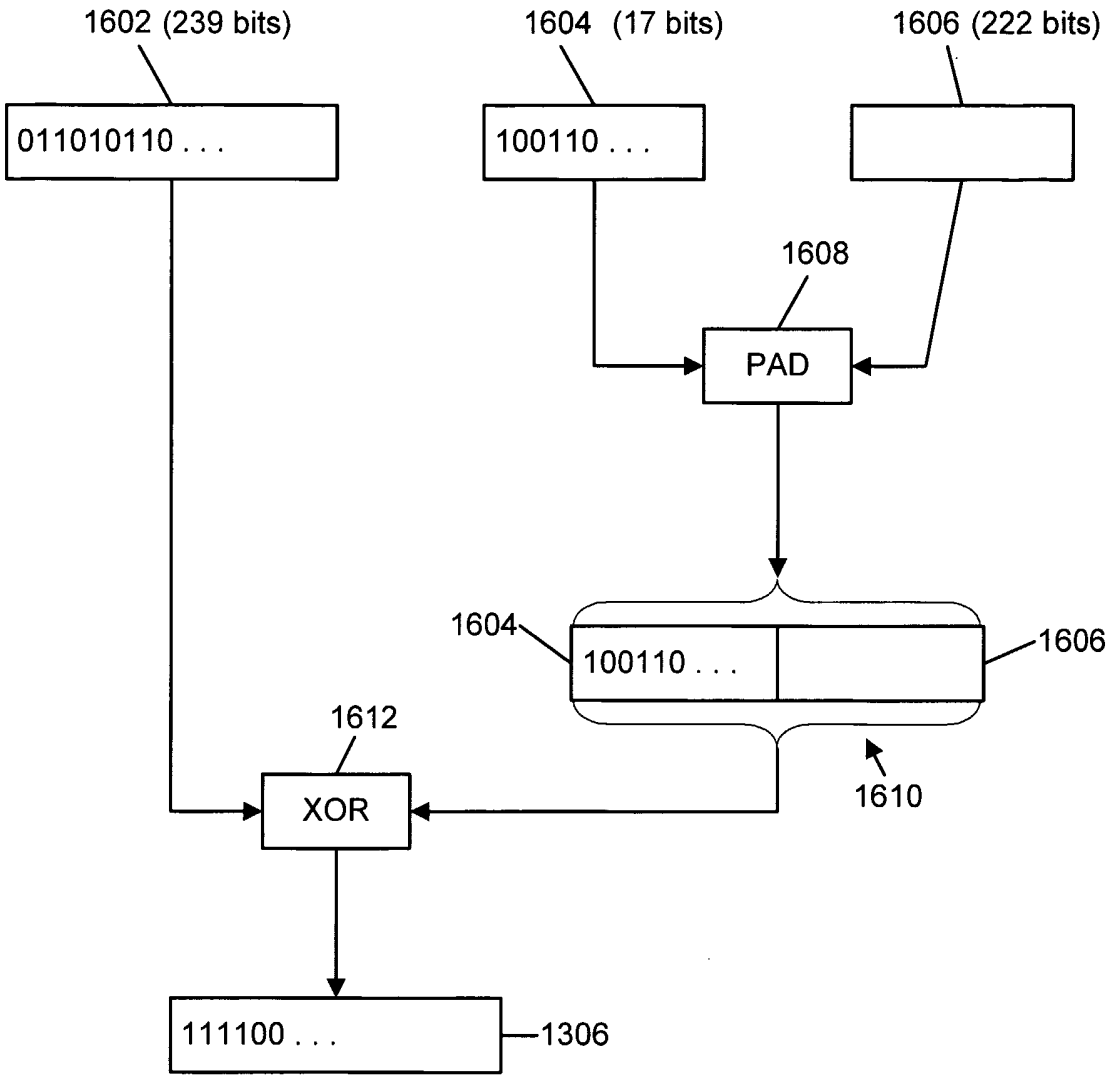

1602 (239 bits)
1604 (17 bits)
1606 (222 bits)
011010110 . . .
100110 . . .
1608
PAD
1604
100110 . . .
1606
1612
XOR
1610
111100 . . .
1306

FIG. 16