Multimedia Systems
DOI 10.1007/s00530-013-0327-z

REGULAR PAPER

A proposal of digital rights management based on incomplete cryptography using invariant Huffman code length feature

Ta Minh Thanh · Munetoshi Iwakiri

Abstract Digital rights management (DRM) system is a promising technique to allow copyrighted content to be commercialized in digital format without the risk of revenue loss due to piracy. However, traditional DRMs are achieved with individual function modules of cryptography and watermarking. Therefore, all digital contents are temporarily disclosed in perfect condition via decryption process in the user-side risking illegal redistribution. This paper describes the basic idea of a novel DRM method composed of an incomplete cryptography using invariant Huffman code length feature and the user identification mechanism to control the quality of digital contents. The proposed incomplete cryptography consists of two processes: the incomplete encoding and the incomplete decoding. These processes are presented by randomly selecting the coefficients that belong to the same category or different category of Huffman code. In our scheme, the copyright information is embedded into the decoded content during the decoding process, and the size of digital contents are invariant during the process. Experimental results with simulation confirmed that the modified codes are compatible with standard JPEG format, and revealed the proposed method to be suitable for DRM in the network distribution system.

Keywords Digital rights management (DRM) · Huffman code · JPEG algorithm · Digital images · Copyright protection · Incomplete cryptography · Invisible watermarking · Invariant Huffman code length feature

T. M. Thanh (✉)
Department of Computer Science, Tokyo Institute of Technology, 2-12-2, Ookayama, Meguro-ku, Tokyo 152-8552, Japan
e-mail: thanhtm@ks.cs.titech.ac.jp; taminhjp@gmail.com

T. M. Thanh
Department of Network Security, Le Quy Don Technical University, 100 Hoang Quoc Viet, Cau Giay, Hanoi, Vietnam

M. Iwakiri
Department of Computer Science, National Defense Academy, 1-10-20, Hashirimizu, Yokosuka-shi, Kanagawa 239-8686, Japan
e-mail: iwak@nda.ac.jp

1 Introduction

1.1 Background

Advances in computer and network technologies have made it easy to copy and distribute the commercially valuable digital content, such as video, music, picture via global digital networks. This enables an e-commerce model consisting of selling and delivering digital versions of content online. The main point of concern for such a business is to prevent illegal redistribution of the delivered content.

Digital rights management (DRM) systems were created to protect and preserve the owner's property right for the purpose to protect their digital contents. A DRM system usually contains encryption and key management, access control, copy control, identification, tracing and billing mechanisms. In general, in order to protect a DRM system against tampering a hardware-based protection is used, often implemented in set-top boxes. However, the biggest disadvantage of hardware-based DRM systems is inflexibility and high cost. It requires a large investment cost from the service provider and increases time to market. Additionally, hardware-based DRM systems are expensive for customers. At a time where a lot of pirated content is available on the

Published online: 14 June 2013

Internet, hardware-based solutions have a hard time creating value for the consumer. In order to reduce the investment cost, the software-based DRM [1–4] is proposed instead of hardware-based DRM. The advantage of software-based DRM is that they can cheaply be distributed to the customers via networks and does not need to create additional installation costs. Most users would prefer a legal way to easily access content without huge initial costs or a long-term commitment. The problem with software-based DRM systems is that they are assumed to be insecure. Especially, such kind of software-based DRM technologies are manipulated by encryption and watermark method separately. Therefore, the original content is disclosed temporarily inside a system in the user's decryption [5]. In that case, users can save original contents without watermark information and distribute via network.

1.2 Our contributions

In this paper, we describe the design and implementation of DRM technique based on an incomplete cryptography system by using the invariant Huffman code length. The incomplete cryptography is proposed for improving the problem of conventional DRM system. Our method will deteriorate the quality of original contents to make trial contents for distribution to widely users via network. The quality of trial contents will be controlled with a watermarked key at the incomplete decoding process, and the user information will be embedded into the incomplete decoded contents simultaneously.

In this study, we do not focus on the robustness of the watermarking method. We concentrate to solve the problem of conventional DRM system which is to completely disclose the original content inside user's system during the decoding process. We combine two processes (the decoding and watermarking) at user side to become the incomplete decoding and the individual user's information is embedded into the decoded content during the decoding process.

1. We proposed the fundamental incomplete cryptography which differs from complete cryptography (e.g., DES, AES,...). It is promising to be able to solve the problem of conventional DRM system.
2. We presented a new DRM system that includes trial contents for advertisement and user ID for distinguishing the legal user. Our system makes it easier for users to try the digital content before deciding whether to purchase it or not.
3. Our proposed method can detect the source of the pirated content by comparing the extracted user ID from the incomplete decoded content with producer's database. It is considered that it can limit the illegal redistribution in advance.
4. We also proposed a new watermarking technique based on the Huffman code length feature in which the size of the digital content is not changed by the whole process.

1.3 Roadmap

This paper is organized as follows. First, the next section presents an overview of related works on DRM applications. Section 3 proposes a fundamental idea of incomplete cryptography. In order to implement our idea to the Huffman codes of JPEG (joint photographic experts group) image, the Huffman encoder and the Huffman decoder algorithms are summarized in Sect. 4. Section 5 presents the incomplete cryptography method based on the invariant Huffman code length of the single Huffman category. With another idea, we also propose the new method of incomplete cryptography by using the combination of multiple Huffman categories to control the length of Huffman code in Sect. 6. The experimental results with JPEG algorithm are given in Sect. 7 and Sect. 8 summarizes the conclusion.

2 Review of related works

Our research has been inspired by a number of conventional works available in the literature that employ digital watermarking for copyright protection of digital content with two targets: digital content access and traitor tracing. We also investigate the techniques that use both encryption and fingerprinting in the literature. In general, server-side watermark embedding and user-side watermark embedding techniques are mainly employed to implement the digital content distribution system. The existing joint multimedia encryption and fingerprinting technologies can be divided into three categories [6]. They are briefly described as follows.

Server-side encryption and watermark embedding scheme [1, 2]: An original content is separately embedded with a user's watermark and then encrypted with a global key to create an encoded content. However, there are some disadvantages in this model: the first one is inefficient bandwidth utilization because digital fingerprint embedding is done at the server, then it will be the high computation cost if the user repeats request of same content many times. The second one is insecure because only a single global encryption key is used. If a malicious user can decode another user's data, then original content can be obtained and be illegally distributed. Therefore, this scheme cannot trace the illegal distributors who send their copies to unauthorized users.

Server-side encryption and user-side watermark embedding (conventional DRM system) [7–11]: Macq

et al. [7] proposed the fingerprint embedding at the users side for digital TV. After that, Hartung et al. [8] and Bloom et al. [9] extended the idea of [7] for applying to DRM in digital cinema. In this model, only one global key encryption is employed to encode the digital content at the server side. Next, the encoded content can be sent to different users via network. At the user side, the encoded content can be decrypted according to the global key and it must to be fingerprinted based on user's information. Meanwhile, the digital fingerprint must be embedded into the content after decoding to generate the fingerprinted content for each user. In general, a watermarked software (DRM controller software) is necessary in this scenario for embedding the fingerprinting information into content after decoding the encrypted content. However, the watermarked software is still an open problem because *the original content is possibly revealed inside the system by this software*.

Joint fingerprinting and decryption (JFD) [6, 12–14]: Kundur et al. [6] proposed a JFD method to reduce system complexity and achieve the real-time requirement. In JFD method, the encoded content is partially decrypted such that the un-decrypted parts imitate multimedia fingerprint embedding. JFD is conceptually promising, achieving partial multimedia decryption and multimedia fingerprint embedding at the same time. However, the un-decrypted parts should not affect the whole transparency of the fingerprinted content and it should be robust against some attacks.

In this paper, we present a new incomplete cryptography method, which can be incorporated into the DRM system. Most importantly, this method does not encounter the same problems as the above three types of methods. In the next section, incomplete cryptography method will be briefly described.

3 The idea of incomplete cryptography

Our proposed incomplete cryptography [15–17] to DRM system is explained in this section. There are two steps in the proposed cryptography: the incomplete encoding and the incomplete decoding.

3.1 Incomplete encoding

The incomplete encoding process is presented in this section. The basic idea of the incomplete encoding algorithm is shown in Fig. 1. Producer T has a digital content P and needs to create an encoded content by the incomplete cryptography. In that case, P will be encoded based on the encoder function E with the encoder key k to make the scrambled content C.

$$C = E(k, P) \tag{1}$$

In this paper, E randomly chooses the DCT coefficient in the same category or the different category based on the Huffman code length feature to encode the original DCT coefficient chosen by k.

In the incomplete cryptography, C can be simply recognized as a part of P (even if C is not decoded). This feature is called *incomplete confidentiality*. T can widely distribute C to users as trial content via network.

3.2 Incomplete decoding

The incomplete decoding process is different from the complete decoding process. The decoded content is created by another decryption function D' and decoded key $k_i'(i = 1, 2, \ldots, n)$. The decoded content P_i' is calculated as follows:

$$P_i' = D'(k_i', C) \tag{2}$$

In this case, because P_i' is decoded by another decryption function D' with key k_i', it will be deferent from original content P. Therefore, the relationship of P and P_i' is $P_i' \neq P$ in the incomplete cryptography system. As shown in the Fig. 1, D' chooses the DCT coefficient in the same category or the different category based on the watermark bit extracted from user ID. Hence, this decoding process is

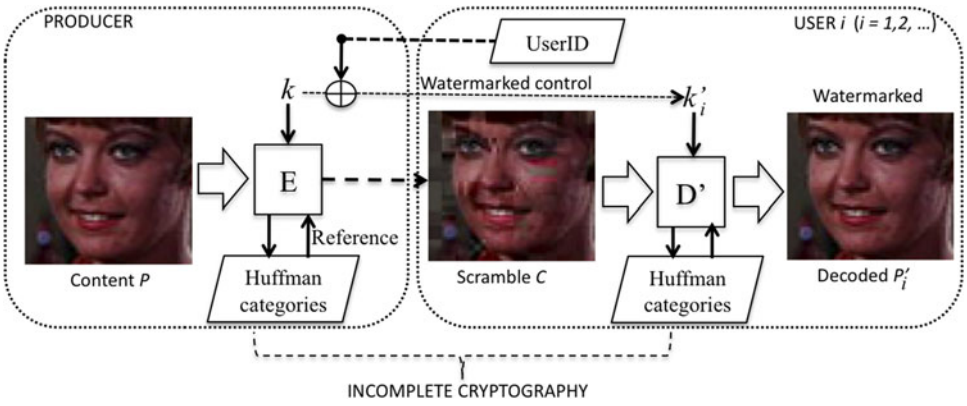


Fig. 1 The overview of incomplete cryptography

quite different from complete cryptography. This feature is called *incomplete decode*.

According to features of the incomplete cryptography, if a set of decoder key $k_i^{'}$ with decoder function $D^{'}$ that are employed to decode an encoded C are chosen, a set of decoder content $P_i^{'}$ will be created and it is different from each other. So, if the incomplete cryptography is implemented to construct a distribution system via network, the producer can distinguish the legal user by $P_i^{'}$ that is decoded based on key $k_i^{'}$.

On the other hand, considering the algorithm of incomplete cryptography, we also propose a novel watermark technique using the incomplete decoding. For watermark embedding, first, the watermarked key is generated to control quality of the content while the decoding process (see Fig. 1). Suppose $k_i^{'}$ is the watermarked key and w_i is the watermark information (user ID). $k_i^{'}$ is expressed as,

$$k_i' = k \oplus w_i \tag{3}$$

As shown in formula (2), when $k_i^{'}$ is used to decode C, w_i will be embedded into $P_i^{'}$ as the copyright information. w_i is useful to confirm the legal users or to trace the source of the pirated copies.

Thus, T can control the quality of $P_i^{'}$ (watermarked contents) with a particular key $k_i^{'}$ (watermarked key). Then, when the user decodes C by using $k_i^{'}$ to achieve $P_i^{'}$, $P_i^{'}$ is not only decoded with slight distortion, but also watermarked with individual user information that is used as watermarking information. It is the elemental mechanism of watermarking based on the incomplete cryptography system.

Note that, in this study, robustness is not the major concern. Therefore, we derive analytic bounds of the embedded signals to achieve the highest transparency and ensure that our technique can trace the traitor exactly.

3.3 DRM system based on incomplete cryptography

The idea of the DRM system (Fig. 2) based on the incomplete cryptography is presented in this subsection. A DRM system is required to enable the distribution of original contents safely and smoothly, as well as to enable the secondary use of contents under rightful consents. When a DRM system is constructed by using incomplete cryptography to implement a content distribution system, it is not only the safe distribution method to users, but also the solution of the conventional DRM problem (disclose the original content inside user's system). The workflow of the proposed DRM system can be explained as follows:

Step 1 T encodes the digital content P to make the scramble content C by using the incomplete encryption function E (Fig. 2a). C is widely distributed via network.

Step 2 R downloads C without the license and tries C to decide whether purchase P or not.

Step 3 After deciding to purchase P, R sends the user ID w_i to T to register the license.

Step 4 T creates the watermarked key $k_i^{'}$ based on w_i and sends $k_i^{'}$ to R. T also saves w_i to user ID database and this database will be used for comparing with the extracted user ID that is retrieved from the suspected content.

Step 5 R decodes C by using $k_i^{'}$ to obtain the watermarked content $P_i^{'}$ (see Fig. 2b).

Step 6 If R illegally redistributes $P_i^{'}$ via another network, T can detect R by tracing the user ID from his/her database.

In the proposed DRM system, the user ID w_i and the watermarked key $k_i^{'}$ are managed by producer T in the server-side. Therefore, when a producer wishes to check whether the user is a legal user, he/she can extract the watermarking information from $P_i^{'}$ and compare with his/her user database. If the watermarking information matches his/her database, the user is a legal user. Conversely, if the watermarking information is different from his/her database, the user is an illegal user. Furthermore, it can specify to trace the source of pirated copies. The purpose of this proposed method is to inform users about the existence of watermarking which can exactly identify users, and limit the illegal redistribution in advance.

3.4 Comparison of our method and conventional method

There are three primary technologies currently used to identify content in the new unstructured distribution: *digital watermarking*, *digital fingerprinting* and *JFD*. While the three enable content identification, they differ in some significant ways that bear on their appropriateness for different applications. Our proposed method differs from those techniques. Table 1 shows the brief comparison of conventional methods with our proposed method.

We compare our method with the conventional methods on a number of criteria to provide a better understanding of the advantages of proposed approach.

- *Embedding information* is defined here as the information embedded into the digital content after distribution to users.
  Watermarking is generally using copyrights information for embedding. Watermarking is not intended to identify the user. The presence of the watermark is extracted to prove that the content is a copy. *Fingerprinting* is generated by content-based processing such as fingerprint of human. Normally, the hash value of digital content is used to distinguish the contents. In

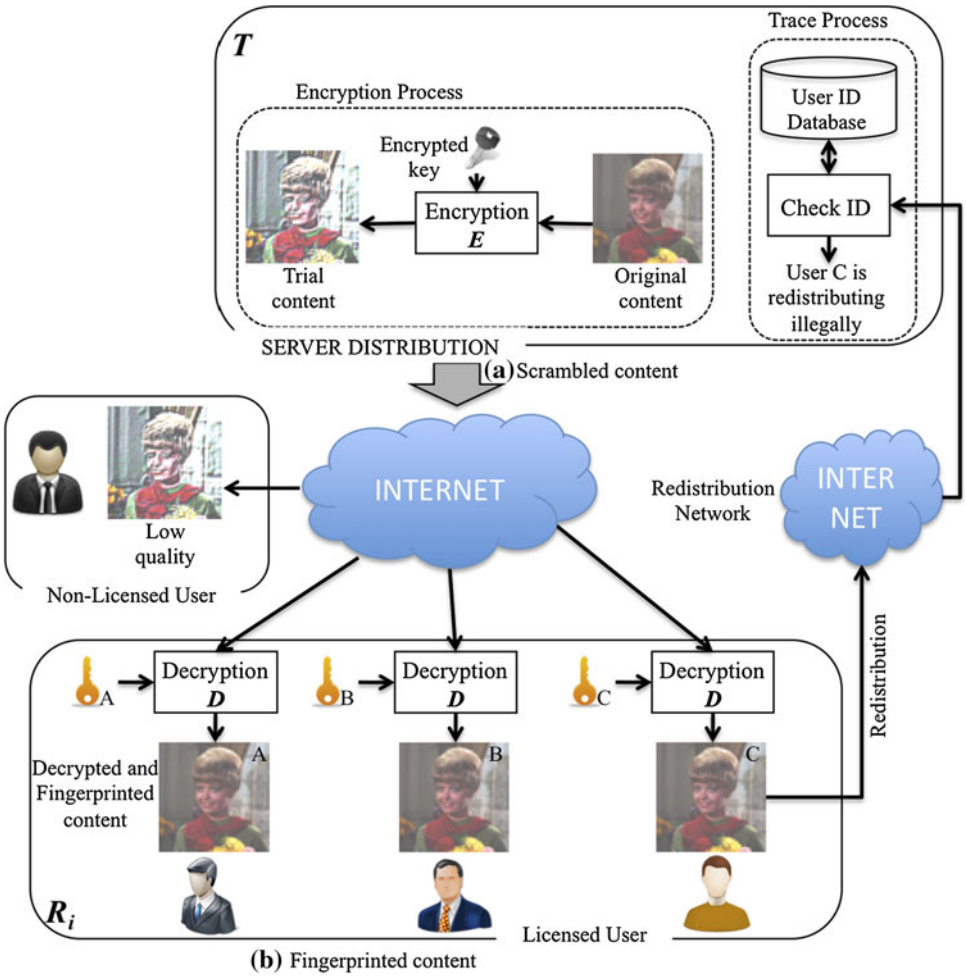


Fig. 2 DRM system based on incomplete cryptography

Table 1 Comparison of exist methods and our method

	Watermarking	Fingerprinting	JFD [6]	Ours
Embedding information	Watermark	Hash	Un-decrypted parts	User ID
Trial content	×	×	◯	◯
Cost	×	×	◯	◯
Illegal user identification	Check watermark	Check hash	Check un-decrypted parts	Check user ID
Traceability	×	Δ	Δ	◯

Note: "×" means Bad; "Δ" means Good; "◯" means Very Good

JFD method, the un-decrypted parts imitate multimedia fingerprint embedding in decoding process. And in *our method*, the user ID is used to embed into the decoded content for individual user when user decodes the trial content.

- *Trial content* is the low quality contents, which is distributed to users widely via network.
  In conventional *watermarking, fingerprinting* method, there is no concept of trial content. The copyrights information is directly embedded into the content before delivering to user. Otherwise, *our method* and *JFD* method employ the trial content for advertisement content. Then, users can try the content beforehand to decide purchasing it or not.
- *Cost* is considered the objective costs like computer resources and more subjective factors such as system complexity.
  In *watermarking, fingerprinting* method, since embedding information is directly embedded into original content before distributing, server-side is almost responsible for embedding by high computation cost. If watermarking and fingerprinting is used for distinguishing the legal users, the user-based information should be embedded into content before distribution. Therefore, the cost computation of these methods are not good. On the other hand, *our method* and *JFD* shift the decoding process to user-side, then high computation cost in server-side is clearly reduced.

- *Illegal user identification* is the measure method to detect whether the user is an illegal user.
  In *watermarking* method, if producer can detect the watermarking information from redistributed content, the user who possessed that content will be judged as illegal user. In *fingerprinting* and *JFD* method, when detected fingerprint does not match against a reference database or un-decrypted parts does not match against database, the content possessor will be judged as illegal user. Otherwise, in *our method*, we define that *when user ID is not detected or does not match against database*, then user is judged as illegal user. Therefore, all attacks that try to remove or replace the user ID from an embedded content will be disabled. And the content, which is generated by those attacks will be considered as illegal content.
- *Traceability* is the ability of system for detecting the source of pirated content.
  Watermarking method is generally used for copyrights protection, then watermark information is not particular user information. So, watermarking cannot trace the source of pirated content. JFD method is considered since it can trace the source of the pirated content by detecting the un-decrypted parts of individual user in decoded content, but it seems quite complicated. Unlike watermarking and JFD method, user ID is employed for information embedding in our proposed method and fingerprinting. Thus, if we detect the user ID from the decoded content, we can exactly trace the illegal source of pirated content.

4 Huffman code in JPEG and the problem of conventional watermarking method

There are several approaches related to Huffman code watermarking [18–22]. These methods were studied with viewpoints of the watermark robustness, it means that the embedded information remains even after attempts to tamper with the image data. And, the other approaches were developed to maintain the high quality even after embedded copyright information into the digital content. However, there is a problem in these methods: the size of the watermarked content is changed for each legal user because the size of the content is determined by the length of Huffman code in the Huffman table [23–25].

4.1 Summary of Huffman code in JPEG algorithm

JPEG is an image compression algorithm and an image file format of international standard, and it is used in global applications now [26].

Images subjected to JPEG encoding are first broken down into 8×8 blocks. Next, each block is put through the discrete cosine transform (DCT), then the DCT coefficients are quantized into integers using a quantization table, and finally entropy encoding is performed. In general, the spectrum of the image is biased toward the lower range, and as a result the DCT coefficients in higher ranges are often set to zero as a result of quantization. The last step in this process is to compress these coefficients using Huffman encoding.

In case of JPEG, image information is kept inside the data file as a quantized DCT coefficient and quantization table. On the other hand, various parameters such as the quantization table coefficients, and side information, which are necessary to decode the picture, are recorded in the frame header. Quantized DCT coefficients are stored in the DCT tables (8×8) by zigzag scanning, where the DC coefficient is the value of the top-left corner [(0, 0) coefficient]. The remaining 63 coefficients are called the AC coefficients. The quantized DCT coefficients, which are in the neighborhood of the DC coefficient, are low-frequency coefficients, and the others correspond to the high-frequency coefficients. Because the high-frequency coefficients in 8×8 block are often become "0" after quantization, the spectrum of picture tends to be constructed with low-frequency coefficients.

This means that the run-length coding is suitable to the high-frequency coefficients of DCT blocks by use of the zigzag scanning. The "0" run-length of AC coefficients is typically longer by the zigzag scanning, and it achieves high efficient compression, i.e., image data size is reduced by the quantization, the zigzag scanning and the run-length encoding based on the Huffman codes.

In the JPEG algorithm, Huffman codes of DC/AC coefficients groups are processed in different methods, respectively.

4.2 Huffman code of DC coefficients

First, we explain the processing of the DC coefficients. Generally, the DC coefficients between adjacent DCT blocks have strong correlation. Therefore, the JPEG encoder takes differential value of the DC coefficients between adjacent blocks. These differential values are encoded using Huffman codes. By this process, the quantity of DC coefficients data is compressed. Table 2 is the Huffman encode table for DC coefficients.

4.2.1 Encoder algorithm

Step 1 Obtain the difference *diff* of the DC coefficients in the previous DCT blocks.
Step 2 Find the category *S* of *diff* in Table 2 and take variable-length codes and additional bits corresponding to *S*.

Table 2 Huffman code table for DC coefficients

diff	*S*	Huffman code	Additional bits
−2047,...,−1024,1024,...,2047	11	111111110	00000000000,...,01111111111,10000000000,...,11111111111
−1023,...,−512,512,...,1024	10	11111110	0000000000,...,0111111111,1000000000,...,1111111111
−511,...,−256,256,...,511	9	1111110	000000000,...,011111111,100000000,...,111111111
−255,...,−128,128,...,255	8	111110	00000000,...,01111111,10000000,...,11111111
−127,...,−64,64,...,127	7	11110	0000000,...,0111111,1000000,...,1111111
−63,...,−32,32,...,63	6	1110	000000,...,011111,100000,...,111111
−31,...,−16,16,...,31	5	110	00000,...01111,10000,...,11111
−15,...,−8,8,...,15	4	101	0000,...,0111,1000,...,1111
−7,...,−4,4,...,7	3	100	000,...,011,100,111
−3,−2,2,3	2	11	00,01,10,11
−1,1	1	10	0,1
0	0	00	Non

Step 3 Coupling the variable-length codes and the additional bits to output the Huffman code of the DC coefficient.

4.2.2 Decoder algorithm

Step 1 Take the Huffman code from the extracted JPEG bitstream and find out a category *S* from Table 2.
Step 2 Detect the difference *diff* of the DC coefficients and the additional bits from *S*.
Step 3 If the most significant bit (MSB) of the additional bits is 0, then the *diff* is negative value. After obtaining *diff* + 1, add bit "1" before least significant bit (LSB) of $(S+1)$th bit and change to negative value.

4.3 Huffman code of AC coefficients

Table 3 is used for the AC coefficients in the encoder/decoder as the Huffman codes.

4.3.1 Encoder algorithm

Step 1 In the zigzag scan, we can obtain the zero length R_c value of the AC coefficients in the DCT block.
Step 2 Find the category *S* of the each nonzero AC coefficient from Table 2.
Step 3 From the combination of R_c and *S* in Table 2, we can take the variable-length codes for the each AC coefficient.
Step 4 Coupling the variable-length codes and the additional bits corresponding to *S* in Table 3, the Huffman code as the AC coefficient is obtained.

4.3.2 Decoder algorithm

Step 1 Take the Huffman code from the extracted JPEG image and obtain R_c, the nonzero AC coefficient by referring Table 3.
Step 2 If it means the ZRL (zero run length), 16 AC coefficients are updated and back to Step 1. It means that the 16 AC coefficients are 0.
Step 3 If it means the EOB (end of block), the decoder of this block finishes. This is equivalent to the all remaining AC coefficients are 0.
Step 4 If *S* is not 0, the additional bits is obtained. If the MSB of the additional bits is 0, the AC coefficient is negative value. After $AC+1$, add bit "1" before least significant bit (LSB) of $(S+1)$th bit and change it to negative value. Repeat this step to finish the decode of DCT block.

According to this algorithm, the DC/AC coefficients in JPEG image are encoded and decoded to the Huffman code. Therefore, after applying the Huffman code algorithm, the size of image is changed as mentioned in [23, 24].

Table 3 Huffman code table for AC coefficients

S	0	1	2	...	10
R_c					
0	1010(EOB)	00	01	...	1111111110000011
1	Non	1100	11011	...	1111111110001000
2	Non	11100	11111001	...	1111111110001110
...	...	...	...	...	...
15	11111111001(ZRL)	1111111111110101	1111111111110110	...	1111111111111110

4.4 The variant file size problem in conventional watermarking method

In general, almost conventional watermarking method is not considered the Huffman code. This is the reason why the size of digital content is changed before and after embedding information into the digital content. For showing the problem of conventional watermarking method, we describe an example of paper [16] in Fig. 3 which is not considered the Huffman code in AC coefficient.

Suppose that Fig. 3a is a part of DCT table in the original JPEG image P. We have the zigzag scanning result as $\{2, 8, -9, 3, EOB\}$. Next, the entropy code of the AC coefficients $\{8, -9, 3, EOB\}$ in this DCT table can be calculated as follows:

Step 1 Create pair (R_c, AC) from AC coefficients.
$\{(0, 8), (0, -9), (0, 3), EOB\}$
Step 2 Obtain category S by AC coefficients.
$\{(0, 4), (0, 4), (0, 2), EOB\}$
Step 3 As in Table 3, we can find out the variable-length codes of pair (R_c, AC).
$\{1011, 1011, 01, 1010\}$
Step 4 Coupling the results in Step 3 with the additional bits in Table 2, we can obtain the Huffman code for the each AC coefficient in Fig.3a.
$\{1000, 0110, 11, 1010\}$.

From Steps 1–4, Huffman code bits of AC coefficients are created,

$\{10111000\ 10110110\ 0111\ 1010\}$

and the bitstream length is 24 bits.

To scramble for making trial content C, P is encrypted and result is shown in Fig. 3b. The zigzag scanning result is $\{22, \underline{10}, \underline{15}, \underline{16}, EOB\}$. As Steps 1–4 above, if these AC coefficients in this table are converted to the Huffman code, the result is

$\{10111010\ 10111111\ 1101010000\ 1010\}$

and the bitstream length is 30 bits.

On the other hand, after decoding, Fig. 3b, c is obtained and it is also the watermarked DCT table. Again, we observe the zigzag scanning result $\{2, \underline{9}, \underline{-9}, \underline{4}, EOB\}$. We calculate the Huffman code of this DCT table as the Step 1–4 above and obtain the Huffman code is,

$\{10111001\ 10110110\ 1000100\ 1010\}$

and the bitstream length is 27 bits.

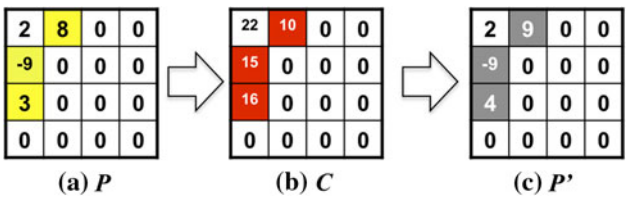


Fig. 3 The problem of watermarking based on incomplete cryptography

According to the results of Huffman code in DCT table of the original image, the scrambled image and the watermarked image, we recognize that the length of Huffman code is variant through processes (24, 30 and 27 bits, respectively) and it is the reason for variant file size because the file size is decided by Huffman code length of each DCT table in JPEG image. Since the size of the watermarked content is changed and it differs for each user, it would be at risk of collusion attack, where multiple attackers (or colluders) perform a linear combination of their decoded contents to result in another content with an objective to confuse the producer so that their individual watermark information cannot be detected properly.

To address this problem, we can use the constant length feature of additional bits for each category S (see Tables 2, 3) to control the scrambled coefficients and watermarked coefficients. Since the coefficients in the same category have the constant length of additional bits, the file size of content is invariant in the whole processes. Besides, we also describe another idea to implement the incomplete cryptography using the invariant offset Huffman code length feature of AC-coefficient which uses multiple categories for controlling the length of Huffman code in DCT table. The detail of algorithm is shown in next section.

5 Single Huffman category-based incomplete cryptography

In this section, we use the invariant Huffman code length feature of single Huffman category to implement the incomplete cryptography and introduce a new watermark scheme which obtains the watermarked content without changing the data size based on the proposed method.

5.1 The management and generation of the encoding and decoding keys

The producer T uses the encode key k with the encode function E to encode the original content P. E is the encode function that randomly selects the encoded coefficient belonging to the same category in Table 2 (in DC coefficient case) or Table 3 (in AC coefficient case). C is distributed to the user R. After trial C, R registers his/her information to purchase the content. T uses R's information as the watermark information w_i, the encode key k, the secret key k_s and the watermarked key generation function G to create the decode key k'_i and send k'_i to R.

While creating the decode key $k_i^{'}$, the method of embedding is presented. The coefficients are selected by $k_i^{'}$ and the embedding position is registered by k_s, namely, if the embedding data bit is "0", the decoded coefficient is the original DCT coefficient (AC or DC). Likewise, if the embedding data bit is "1", the decoded coefficient is a DCT coefficient which belongs to same category of original DCT coefficient.

In the decoding process, R uses $k_i^{'}$ with the decode function $D^{'}$ to decode C and obtains the high-quality watermarked content $P_i^{'}$.

Since $D^{'}$ selects the decoded coefficient belonging to the same category of the encoded coefficient, the size of P, C and $P_i^{'}$ are not changed for the whole progress, and we can obtain the high-quality decoded contents. In addition, using the secret key k_s, the watermark information w_m can be extracted by investigating the specific categories of $P_i^{'}$. Note that, k_s is a key to determine the embedding positions.

In this method, in order to confirm that the user is legal or not, the producer who holds the secret key k_s will extract w_m from $P_i^{'}$ and compare to w_i of the user ID database.

5.2 Invariant Huffman code length AC coefficient Fingerprinting (IHAF)

5.2.1 Algorithm

We employ the feature of Huffman code of AC category, which is invariant for every AC coefficients belonging the same category (Table 3). If the encoded AC coefficient and the decoded AC coefficient are the same category of original AC coefficient, it is expected that the file size is not changed for the whole process.

We choose the AC coefficient p^S belonging category S from P. To create C, p^S is replaced by $p^{'S}$ belonging the same category S. In order to decode C, $p^{'S}$ is decoded according to the watermark bit w. If $w = 0$, $p^{'S}$ is replaced by original AC coefficient p^S. Otherwise, $p^{'S}$ is replaced by another AC coefficient $p^{''S}$ that is near p^S.

5.2.2 Explanation of IHAF method

An example of implementation to the incomplete cryptography using the variant Huffman code of AC coefficient is shown in Fig. 4. Figure 4a is a part of DCT table in P. We have the zigzag scanning result as {2, 1, 3, 3, *EOB*}. Next, the entropy code of the AC coefficients {1, 3, 3, *EOB*} in this DCT table can be calculated as follows,

Step 1 Create pair (R_c, AC) from the AC coefficients.
{(0, 1), (0, 3), (0, 3), *EOB*}
Step 2 Obtain category S by the AC coefficients.
{(0, 1), (0, 2), (0, 2), *EOB*}
Step 3 As Table 3, we can find out the variable-length codes of the pair (R_c, AC).
{00, 01, 01, 1010}
Step 4 Coupling the results in Step 3 with the additional bits in Table 2, we can obtain the Huffman code for the each AC coefficient in Fig. 4a.
{001, 0111, 0111, 1010}

From Steps 1–4, the Huffman codes of the AC coefficients are taken,

{001 0111 0111 1010}

and the bitstream length is 15 bits.

On the other hand, suppose that Fig. 4a is encoded by E and becomes Fig. 4b. The zigzag scanning result is $\{2, 1, \underline{-2}, 3, EOB\}$. As Steps 1–4 above, if these AC coefficients in this table are converted to the Huffman code, the result is

{001 0101 0111 1010}

and the bitstream length is 15 bits.

Furthermore, after decoding Fig. 4b, c is obtained and it is also the watermarked DCT table. In this decoder process, D selected the decoder coefficient "2" (Fig. 4c), which is in the same category with the original coefficient "3", to decode the encoded coefficient "−2" in Fig. 4b. Again, we have zigzag scanning result of Fig. 4c is $\{2, 1, \underline{2}, 3, EOB\}$. The Huffman code of this table is,

{001 0110 0111 1010}

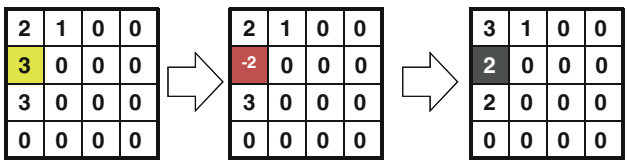
and its length is also 15 bits. In this case, bit $w = 1$ is embedded into DCT table.

As from above the results, the length Huffman code of P, C, $P^{'}$ is also 15 bits. Therefore, we can use IHAF method to implement DRM system based on the incomplete cryptography.

5.3 Invariant Huffman code length DC coefficient fingerprinting (IHDF)

5.3.1 Algorithm

We also use the invariant length code feature of Huffman code for *diff* of DC coefficient, that belongs the same category S (Table 2). We calculate $diff^S$ of ith and $(i + 1)$th DCT



(a) *P* **(b) *C*** **(c) *P'***

Fig. 4 Example of IHAF method

table. In order to create C, $diff^S$ is replaced by $diff'^S$ belonging the same category S. The decoded content P' is created based on watermark bit w. If $w = 0$, $diff'^S$ is restored to $diff^S$. And if $w = 1$, $diff'^S$ is replaced by $diff''^S$ that is near $diff^S$.

5.3.2 Explanation of IHDF method

Figure 5 shows an example of the implementation on IHDF method. Suppose Fig. 5a is a part of ith and $(i + 1)$th DCT table in JPEG image. The ith DCT table has been already encoded and the DC coefficient is "9". Next, the DC coefficient in the $(i + 1)$th DCT table is encoded to Huffman code. This procedure is described as follows.

Step 1 Calculate the difference *diff* of the DC coefficient from the ith and $(i + 1)$th DCT table.

$$diff = 3 - 9 = -6$$

Step 2 Obtain the category S of *diff* by refer Table 2. In this example, $S = 3$. Therefore, the variable-length bit is {100} and additional bit is {001}.

Step 3 Coupling the results in Step 2, we can create the Huffman code of the DC coefficient in the $(i + 1)$th DCT table. It is {100 001}.

According to Steps 1–3, the Huffman code of the DC coefficient in the $(i + 1)$th DCT table is obtained and its bitstream length is 6 bits in this example.

When the process of using the $(i + 1)$th DCT table, the DC coefficient is encoded and decoded by *diff* which is selected in the scope of category S. These processes are shown in Fig. 5b, c. In the encryption result (Fig. 5b), original DC coefficient "3" is replaced by "5". Therefore, its Huffman code is {100011}. On the other hand, after decoding, the encoded coefficient "5" is replaced by "4" (Fig. 5c). We recognized that the difference *diff* of DC coefficient in ith and $(i + 1)$th DCT table is not changed for the whole processes, only the additional bit is changed. In the result of the decoding process, the Huffman code of DC coefficient is {100010}.

From the above results, we recognized that the size of P, C, P' is not changed and it also is 6 bits. In this example, $w = 1$ is embedded into DCT table. Consequently, we can implement the DRM system based on IHDF method.

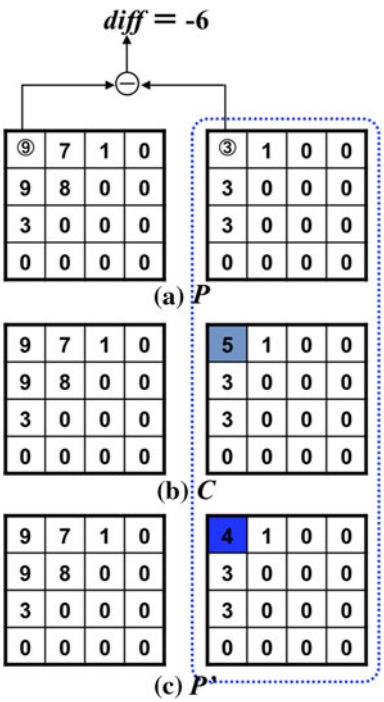


Fig. 5 Example of IHDF method

6 Multiple Huffman categories-based incomplete cryptography

This section presents the new idea by using the combination of multiple Huffman categories to control the length of Huffman code. This idea can be used by using combination of two methods explained in Sect. 5. According to Table 3, the Huffman code length of AC coefficient is variant when category S is changed. Using this feature, our proposed method is conducted on multiple categories and successfully to control the length of Huffman code. Our proposed invariant Huffman code length offset AC coefficient fingerprinting (IHOF) method is explained as follows.

6.1 Algorithm

A pair of AC coefficient $\{p_1^S, p_2^S\}$ of P, which belongs to the same category S in Huffman table, is chosen for encryption. E is used to encode $\{p_1^S, p_2^S\}$ by replacing with $\{p_1^{S+M}, p_2^{S-M}\}$ that belongs to category $S + M$ and $S - M$, respectively. After encoding, C is distributed widely via network. P' is also created based on the watermark bit w. If $w = 0$, the pair $\{p_1^{S+M}, p_2^{S-M}\}$ is restored to original value $\{p_1^S, p_2^S\}$. Otherwise, $\{p_1^{S+M}, p_2^{S-M}\}$ is replaced by $\{p_1'^S, p_2'^S\}$ that belongs to the same category S.

6.2 Explanation of IHOF method

An example of implementation to the incomplete cryptography using IHOF method is shown in Fig. 6. Firstly,

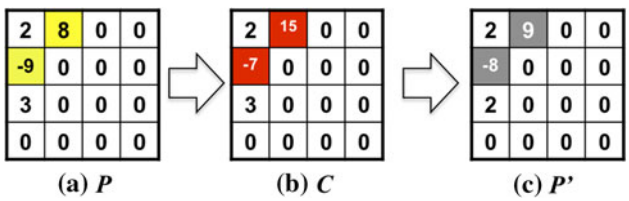


Fig. 6 Example of IHOF method

Fig. 7 The experimental SIDBA images. *Top row* (*left* to *right*) Lighthouse, Pepper, Title, Lenna, Girl. *Middle row* (*left* to *right*) Airplane, Parrots, Couple, Milkdrop, Mandrill. *Bottom row* (*left* to *right*) Earth, Sailboat, Balloon, Aerial, Watermark logo

Table 4 IHAF method: PSNR (dB)/size (bytes) and embedded bits

	P	C	P'	w_m (bits)
Airplane	30.20/13,112	23.29/13,112	30.19/13,112	1,254
Girl	32.70/9,947	27.18/9,947	32.69/9,947	506
Parrots	34.25/10,602	25.57/10,602	34.24/10,602	604
Couple	34.06/9,930	27.48/9,930	34.04/9,930	490
Title	31.84/23,716	15.17/23,716	31.82/23,716	4,566
Lenna	32.37/12,610	24.37/12,610	32.36/12,610	1,012
Milkdrop	31.99/10,894	23.88/10,894	31.97/10,894	822
Mandrill	24.97/23,156	21.63/23,156	24.96/23,156	900
Lighthouse	32.66/14,042	23.02/14,042	32.65/14,042	1,326
Pepper	28.81/13,940	22.57/13,940	28.80/13,940	1,224
Balloon	34.92/8,221	30.43/8,221	34.91/8,221	202
Aerial	28.25/17,890	24.27/17,890	28.24/17,890	798
Sailboat	30.98/14,420	22.65/14,420	30.97/14,420	1,464
Earth	33.66/13,202	25.61/13,202	33.63/13,202	964

we calculate the Huffman code length of original DCT table (Fig. 6a). We have the zigzag scanning result as $\{2, 8, -9, 3, EOB\}$. Only the AC coefficients of this DCT table is changed to Huffman code,

{10111000 10110110 0111 1010}

and the bitstream length is 24 bits.

Here, we recognize that there are two AC coefficients, which are "8" and "9", belong to category $S = 4$. So that, in this example, one pair of AC coefficient is selected to explain our proposed method in detail. According to our algorithm, two coefficients of that pair are encoded by another coefficients in the difference category S but it is needed to control the Huffman code length. On the other

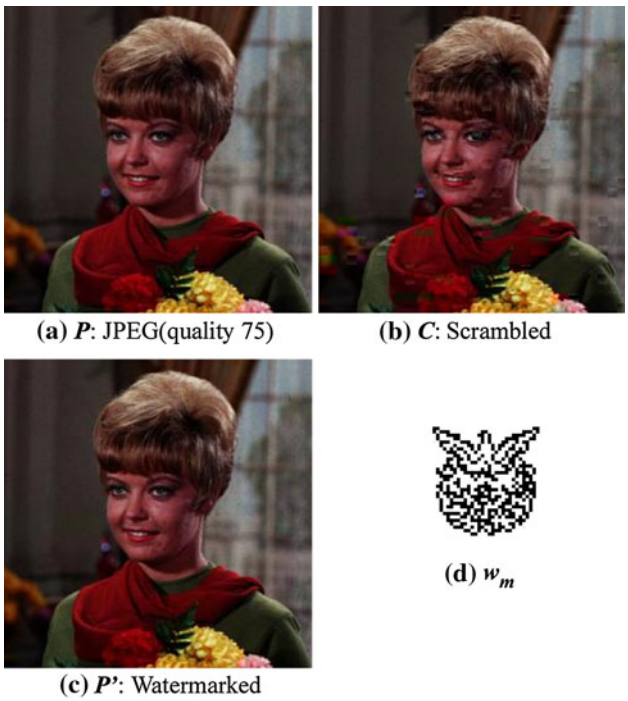


Fig. 8 Example of Girl image in IHAF method

hand, the encoded pair is decoded by AC coefficient in the same category $S = 4$ depending on the watermark bit.

In complete encoding process, suppose that encoded key k randomly selects "16" to encode "8" in above pair. It means that, the original AC coefficient "8" in category $S = 4$ is changed to coefficient "16" in category $S = 5$. Therefore, the Huffman code length of encoded DCT table is different from original DCT table. To ensure the Huffman code length of encoded DCT table is the same as original DCT table, k should select the coefficient in category $S = 3$ to encode the remaining coefficient "−9". In example Fig. 6b, "−7" is chosen to replace original coefficient "−9". In this case, (8, −9) pair is encoded to (16, −7). Again, we calculate the Huffman code length of the encoded table Fig. 6b to confirm the efficiency of proposed method. The zigzag scanning of Fig. 6b is $\{2, \underline{16, -7}, 3, EOB\}$. Note that, we only encoded the AC coefficients and Huffman code of this table is,

$\{\underline{1101}0000\ \underline{100}000\ 0111\ 1010\}$

and the bitstream length is $\underline{24\,\text{bits}}$.

On the other hand, in incomplete decoding process, encoded pair is completely decoded or does not depend on the watermark bit. In the example Fig. 6c, pair (16, −7) is decoded by (9, −8) which are the coefficients belonging to category $S = 4$. Since "9" and "−8" are not original coefficients, it means that bit "1" is embedded into decoded DCT table in decoding process. To check the Huffman code length of this table, we extract the zigzag scanning results as $\{2, 9, \underline{-8}, 3, EOB\}$ and calculate the Huffman code,

$\{\underline{1011}1100\ \underline{1011}0111\ 0111\ 1010\}$

and the bitstream length is $\underline{24\text{bits}}$.

As in above the results, the length Huffman code of P, C, $P^{'}$ is also 24 bits. Therefore, we can use the invariant Huffman code length feature of the offset AC coefficient in difference category S to the implement DRM system based on the incomplete cryptography.

7 Experimental results and evaluation

In this section, we implemented the proposed DRM method using the Huffman code in the incomplete cryptography.

Table 5 IHDF method: PSNR (dB)/size (bytes) and embedded bits

	P	C	P'	w_m (bits)
Airplane	30.20/13,112	27.87/13,112	30.19/13,112	120
Girl	32.70/9,947	28.07/9,947	32.69/9,947	252
Parrots	34.25/10,602	29.34/10,602	34.24/10,602	169
Couple	34.06/9,930	29.87/9,930	34.05/9,930	179
Title	31.84/23,716	27.65/23,716	31.83/23,716	187
Lenna	32.37/12,610	28.53/12,610	32.34/12,610	188
Milkdrop	31.99/10,894	29.15/10,894	31.97/10,894	107
Mandrill	24.97/23,156	23.23/23,156	24.96/23,156	261
Lighthouse	32.66/14,042	29.56/14,042	32.64/14,042	114
Pepper	28.81/13,940	26.83/13,940	28.80/13,940	198
Balloon	34.92/8,221	27.58/8,221	34.87/8,221	229
Aerial	28.25/17,890	25.58/17,890	28.24/17,890	276
Sailboat	30.98/14,420	27.50/14,420	30.96/14,420	195
Earth	33.66/13,202	28.06/13,202	33.62/13,202	205

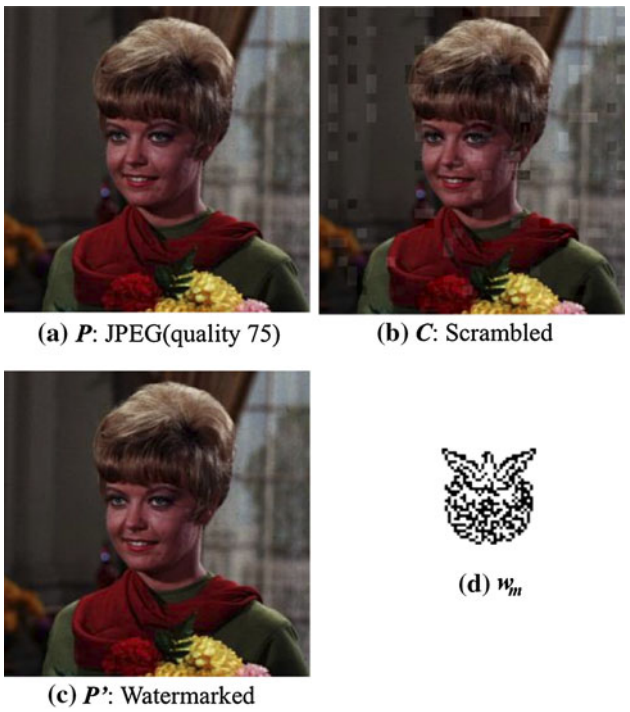


Fig. 9 Example of Girl image in IHDF method

7.1 Experimental environment

In our experiments, all experiments were performed by the incomplete encoding and the incomplete decoding on JPEG images. We use the Vine Linux 3.2 system to perform the experimental system. In order to generate the encryption k, we use function *rand*() of GCC version 3.3.2[1] with $seed = 1$. Additionally, the ImageMagick version 6.6.3-0[2] is used to convert and view the experimental JPEG image.

7.2 Experimental images

We prepared some of different features of the experimental images regarding computer graphics (CG), scenery, construction and person. The 10 test images are 256 × 256 pixels, 8 bit RGB image of SIDBA (standard image data base) international standard images (Fig. 7).

Here, all images are compressed with quality of 75 (the lowest $0 \leftrightarrow 100$ the highest) to make the experimental JPEG images for appraisal of proposed method.

On the other hand, we prepared a bitstream 32 × 32 pixels of binary picture (Nda32) as the watermarking information (w_i) (see Fig. 7).

[1] http://gcc.gnu.org/.

[2] http://www.imagemagick.org/script/.

7.3 Evaluation of image quality

We used PSNR (peak signal-to-noise ratio) [25] to evaluate the JPEG image quality. The PSNR of $M \times N$ pixels images of $g(i, j)$ and $g'(i, j)$ is calculated with

$$PSNR = 20\log\frac{255}{MSE} \quad [\text{dB}]$$
$$MSE = \sqrt{\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\{g(i,j) - g\prime(i,j)\}^2} \quad (4)$$
$$(\text{MSE : Mean Square Error}).$$

In these experiments, the PSNR were calculated with RGB pixel data of original image and the JPEG image. A typical value for PSNR in a JPEG image (quality 75) is about 30 dB [25]. And according to the MOS (mean opinion score) experiment in [16], we realized that the tester felt the deterioration when PSNR of image was lower than approximately 22 dB (MOS 0–2.5). In addition, when PSNR was between 22 and 29 dB (MOS: 2.5–3.5), the tester felt the deterioration but slightly annoying, and the image quality in this case is considered acceptable for the scrambled content. When PSNR is higher than 29 dB (MOS 3.5–5), the testers almost could not feel the deterioration of image. We concluded that PSNR of scrambled content is appropriately between 22 and 29 dB, and PSNR of incomplete decoding should be higher than 29 dB.

7.4 Experimental results of IHAF method

In this experiment, we implement the incomplete cryptography by using IHAF method. We choose the category $S = 4$ for encoding and decoding AC coefficients in DCT table. The original AC coefficient in P is replaced by random coefficient in the category $S = 4$ to make the trial content C. The watermark bit w_i will be embedded by k_i' while decoding C to P'. This means that, the encoded AC coefficient will be decoded with the original AC coefficient in case of the embedding bit is "0". Otherwise, the encoded AC coefficient will be replaced with the another AC coefficient in the category $S = 4$.

We calculated PSNR value of the output JPEG images in every processes and extracted the watermark information (embedded binary data) perfectly from the incomplete decode JPEG images. The results are shown in Table 4 and Fig. 8. From this results, we can see that the watermarked JPEG images P' are not distinguishable from the original JPEG images P. The scrambled JPEG images C are degraded about 20 dB, and they seem appropriate as trial content. The CG image, e.g., Title, seems to be of the worst quality for the trial image because there are numerous of AC coefficients belonging to the category $S = 4$; hence, the quality of image is strongly degraded. However, some of images belong to the person image, e.g., Balloon, does

Table 6 IHOF method: PSNR (dB)/size (bytes) and embedded bits

	P	C	P'	w_m (bits)
Airplane	30.20/13,112	23.30/13,112	30.19/13,112	1,254
Girl	32.71/9,947	27.18/9,947	32.70/9,947	506
Parrots	34.25/10,602	25.57/10,602	34.24/10,602	604
Couple	34.06/9,930	27.48/9,930	34.05/9,930	490
Title	31.84/23,716	15.17/23,716	31.82/23,716	4,566
Lenna	32.37/12,610	24.37/12,610	32.36/12,610	1,012
Milkdrop	31.99/10,894	23.88/10,894	31.97/10,894	822
Mandrill	24.97/23,156	21.63/23,156	24.96/23,156	900
Lighthouse	32.67/14,042	23.02/14,042	32.66/14,042	1,326
Pepper	28.81/13,940	22.56/13,940	28.80/13,940	1,224

not seems to be suitable for the trial image since the quality of trial image is too high. Therefore, in order to create the trial content, producer needs to consider the number of AC coefficient belonging to the category S if the IHAF method is employed.

Moreover, we can confirm that the size of P, C, P' is the same from the experimental results in Table 4. Therefore, IHAF method successfully controls the Huffman code length to maintain the size of content.

7.5 Experimental results of IHDF method

In the implementation of the IHDF method, we also choose the category $S = 4$ for selecting *diff* in the encoding/decoding process. The difference *diff* of DC coefficients is randomly replaced by the another values in the category $S = 4$ to make the encoded content C. In the decoding process, *diff* in C will be replaced with the original *diff* for the embedding bit "0", or with another *diff* in the category $S = 4$ for the embedding bit "1". This decoding process is controlled when R uses k'. Note that, the embedding position in P' will be registered into k_s . k_s is used when T extract the watermarked information from P'.

The experimental results are shown in Table 5 and Fig. 9. From this results, we can see that C is suitable for the trial content with a little bit of high quality. In order to reduce the quality of the trial image, a larger of number of *diff* belong to S needs to be chosen to implement IHDF method. After decoding, we can obtain the high-quality watermarked content P'. The size of P, C and P' is not changed because the IHDF method is used on only the one category S for the encoding/decoding. Otherwise, we can extract w_m (Fig. 9d) from $P^{'}$ after using the secret key k_s and we can compare w_m with w_i to confirm the legal user.

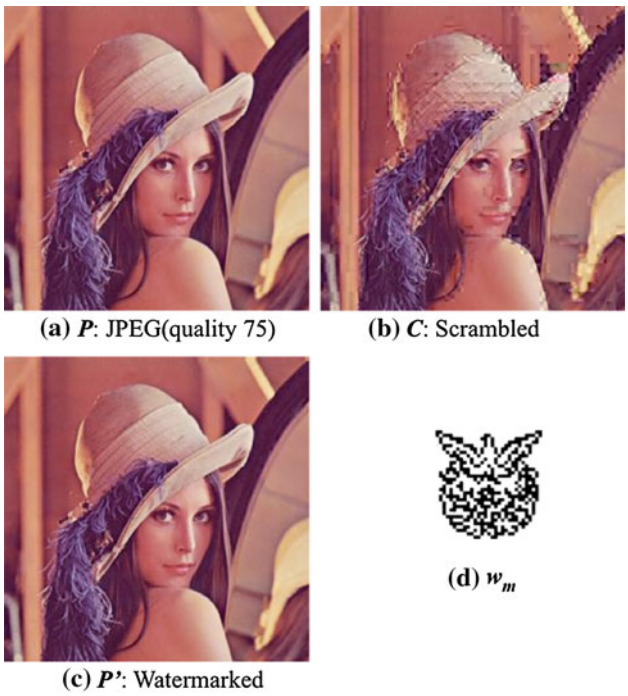


Fig. 10 Example of Lenna image in IHOF method

7.6 Experimental results of IHOF method

In the implementation in IHOF method, we choose two coefficients of category $S = 4$ in each DCT table. To make scrambled content, we encode these two coefficients by another coefficients that belongs to category $S = 3$, $S = 5$ and $M = 1$, respectively. In the incomplete decoding, the encoded coefficients is replaced with the original coefficient for embedding bit "0", or with another coefficient in the same category $S = 4$ for embedding bit "1". Note that, the embedding position in P' will be registered into k_s. k_s is used when T extract the watermarked information from P'.

Table 6 and Fig. 10 show the experimental results of IHOF method. By observing the results, we confirmed that C is suitable for the trial content. After decoding, we can obtain the high-quality watermarked content P'. The size of P, C and P' is not changed for the whole encoding/decoding process. In addition, with the simulation results, we can obtain high quality of decoded content

(fingerprinted content) P' (almost the same as original content P). Otherwise, we can extract w_m (Fig. 10d) from P' after using the secret key k_s and we can compare w_m with w_i to confirm the legal user. By IHOF, we can use the combination of many categories to control the quality of the trial content and the watermarked image.

Based on simulation results, we have established the incomplete cryptography system based on the proposed method. Encoded content (scrambled content) is created to disclose the original content and distributed widely to users. In the incomplete decoding process, we changed the quantized DCT coefficient itself instead of the original quantized DCT coefficient or another DCT coefficient which depends on the category S of Huffman table by a devised decryption key. Thus, the original content is not decoded temporarily inside the system. Therefore, we conclude that the above technical problem by the conventional DRM system is solved by using the incomplete cryptography system.

8 Conclusion

We have proposed an algorithm of the incomplete cryptography for DRM system, which naturally extends that of [16, 17] and employed the Huffman code to implement the incomplete encoding and the incomplete decoding process. We also have proposed a new watermarking technique in which the size of the digital content is not changed by the whole process. According to our proposed algorithm, disclosing the original content problem is solved using the encoder/decoder in the incomplete cryptography. The simulation results indicate that the proposed incomplete cryptography performs well for DRM system. The proposed IHAF, IHDF and IHOF methods show that we can obtain suitable quality of trial content and the high quality of the watermarked content. In addition, the newly proposed method IHOF, which uses the invariant offset Huffman code length feature of multiple AC coefficient categories instead of the invariant code length feature of only one DC/AC coefficient category in [17], results in a flexible algorithm to implement the DRM system by using the combination of multiple Huffman categories.The watermark information extracted from the watermarked content can be used to compare with the user database for tracing the illegal distributor. Therefore, we conclude that the proposed method is useful to the rights management technology under the situation of illegal content distribution via network.

In the future works, our method can be applied on the compression digital content format using Huffman code such as MPEG, H.264 and so on. In addition, our proposal can be extended to other areas, such as multimedia streaming video.

References

1. Kirovski, D., Peinado, M., Petitcolas, F.A.P.: Digital rights management for digital cinema. International Symposium on Optical Science and Technology, Security in Imaging, San Diego (2001)
2. Emmanuel, S., Kankanhalli, M.S.: A digital rights management scheme for broadcast video. Multimed. Syst. **8**, 444–458 (2003)
3. Chang, H., Atallah, M.J.: Protecting software code by guards. In DRM: ACM CCS-8 workshop on security and privacy in digital rights management, pp. 160–175. Springer-Verlag, Berlin (2002)
4. Seki, A., Kameyama, W.: A proposal on open DRM system coping with both benefits of rights-holders and users. IEEE Conf. Image Proc. **7**, 4111–4115 (2003)
5. DRM technology.: Advanced image seminar 2003, The institute of image electronics engineers of Japan (2003)
6. Kundur, D., Karthik, K.: Video fingerprinting and encryption principles for digital rights management. Proc. IEEE **92**(6), 918–932 (2004)
7. Macq, B.M., Quisquater, J.J.: Cryptology for digital TV broadcasting. Proc. IEEE **83**(6), 944–957 (1995)
8. Hartung, F., Girod, B.: Digital watermarking of MPEG-2 coded video in the bitstream domain. Proc. IEEE Intern. Conf. Acoust. Speech Signal Proc. **4**, 2621–2624 (1997)
9. Bloom, J.: Security and rights management in digital cinema. Proc. IEEE Intern. Conf. Acoust. Speech Signal Proc. **4**, 712–715 (2003)
10. Celik, M., Lemma, A., Katzenbeisser, S., van der Veen, M.: Look-up table based secure client-side embedding for spread-spectrum watermarks. IEEE Trans. Inf. Forensic Secur. **3**(3), 475–487 (2008)
11. Piva, A., Bianchi, T., De Rosa, A.: Secure client-side ST-DM watermark embedding. IEEE Trans. Info. For. Sec. **5**(1), 13–26 (2010)
12. Karthik, K., Hatzinakos, D.: Decryption key design for joint fingerprinting and decryption in the sign bit plane for multicast content protection. I. J. Netw. Secur.**4**(3), 254–265 (2007)
13. Lian, S., Liu, Z., Ren, Z., Wang, H.: Secure distribution scheme for compressed data streams, IEEE Conf. on Image Processing (ICIP 2006), Atlanta, (2006)
14. Lemma, A.N., Katzenbeisser, S., Celik, M.U., Veen, M.V.: Secure watermark embedding through partial encryption. In: Proceeding of International Workshop on Digital Watermarking (IWDW 2006), vol. 4283, pp. 433–445. Springer LNCS (2006)
15. Iwakiri, M., Thanh, T.M.: The digital watermark technique by the incomplete cryptography system. Symposium on Cryptography and Information Security 2005 (SCIS2005), 3C1-2, pp. 1039–1044, (2005) (in Japanese)
16. Iwakiri, M., Thanh, T.M.: Fundamental incomplete cryptography method to digital rights management based on JPEG lossy compression, pp. 755–762. The 26th IEEE International Conference on Advanced Information Networking and Applications (AINA-2012), Japan (2012)
17. Iwakiri, M., Thanh, T.: Incomplete cryptography method using invariant Huffman code length to digital rights management, pp. 763–770. The 26th IEEE International Conference on Advanced Information Networking and Applications (AINA-2012), Japan (2012)
18. Noguchi, Y., Kobayashi, H., Kiya, H.: A method of extracting embedded binary data from JPEG bitstreams using standard JPEG decoder. IEICE Trans. Fundam. **E83-A**(8), 1582–1588 (2000)
19. Kiya, H., Noguchi, Y., Takagi, A., Kobayashi, H.: A method of inserting binary data into MPEG video in the compressed domain. IEICE Trans. Fundam.**E82-A**(8), 1485–1492 (1999)

20. Kobayashi, H., Noguchi, Y., Kiya, H.: A method of embedding binary data into JPEG bitstreams. IEICE Trans. **J83-D-II**(6), 1469–1476 (1999)
21. Fridrich, J., Goljan, M., Hogea, D.: New methodology for breaking steganographic techniques for JPEGs. Proc. SPIE **5020**, 143–155 (2003)
22. Kiya, H.: A method of embedding binary data into JPEG. IEICE Trans. **J83-D-II**(6), 1469–1476 (1999)
23. Katzenbeisser, S., Petitcolas, F.A.P.: Information hiding technique for steganography and digital watermarking. Artech House (2000)
24. Cox, I.J., Bloom, J.A., Miller, M.L.: Digital watermarking. Morgan Kaufmann Publishers, San Francisco (1999)
25. Matsui, K.: Fundamentals of digital watermarking. Morikita-publisher, (1998) (in Japanese)
26. The International Telegraph and Telephone Consultative Committee Information Technology-Digital Compression and Coding of Continuous-tone still Images-Requirements and Guidelines, International Telecommunication Union (1992)