Applied Informatics
a SpringerOpen Journal

RESEARCH Open Access

Fragile watermarking based on incomplete cryptography for copyright protection

Munetoshi Iwakiri[1] and Ta Minh Thanh[2,3]*

*Correspondence:
taminhjp@gmail.com
[2]Department of Network Security, Le Quy Don Technical University, 236 Hoang Quoc Viet, Cau Giay, Hanoi, Vietnam
[3]Department of Computer Science, Tokyo Institute of Technology, 2-12-2, Ookayama, Meguro-ku, Tokyo 152-8552, Japan
Full list of author information is available at the end of the article

Abstract

In this work, a fragile watermarking method based on incomplete cryptography for copyright protection is proposed. Our proposed method solves the original content leakage problem of the conventional digital rights management (DRM) system, that is, the original contents are temporarily disclosed inside the user's system. We include the fragile watermarking into the encryption and the decryption of the digital content distribution system via network. Our method can easily control the quality of original content in order to generate the scrambled content for trial before purchasing. Then, when the watermarked content is generated, the user information is simultaneously embedded into the scrambled content. Experimental results with simulation confirmed that our proposed method is successfully applied on the standard JPEG format and it seems to be suitable for real applications.

Keywords: Digital rights management (DRM); Digital images; Copyright protection; Incomplete cryptography; Invisible watermarking

Background

Overview

Since growth of computers and network technology, digital contents are easily manipulated by everyone based on the image/video software. Therefore, illegal copying and distribution of contents via Internet has become a serious problem. Thus, the need for an effective rights management system where only legitimate consumers can have access to digital content, is required recently.

By the traditional protection way, encryption techniques play the role of the first defense method (Shi and Bhargava 1998; Sun et al. 2006; Wen et al. 2002; Xu et al. 2004; Zhu et al. 2005). However, the content after decryption can be redistributed without the permission. It causes illegal distribution even by the legal users. Therefore, the multimedia watermarking and fingerprinting are the potential solutions to the problem (Cox et al. 1999; Katzenbeisser and Petitcolas 2000; Lu et al. 2006a; Wu et al. 2003) by providing passive protection.

Multimedia fingerprinting techniques (Boneh and Shaw 1998; Trappe et al. 2003; Wu et al. 2003; Zhao and Liu 2006) are developed to deal with the illegal redistribution problem. Before transmitting digital content from the content producer to the legal users, a digital fingerprint should have already been embedded into the content. For different users, the corresponding different fingerprints would be embedded into digital content. Once a user illegally redistributes the digital content, the embedded digital fingerprint can be detected to reveal the traitor.

Digital rights management (DRM) systems are created to protect and preserve the owner's property right. A DRM system usually contains encryption, key management, legal access control, and identification of legal user process. To prove the authorized digital content, the watermarking is usually embedded into the content without the knowledge of users. The identification and tracing can be used to follow the source of pirated copies by using the watermarking/fingerprinting (Emmanuel and Kankanhalli 2003; Hartung and Ramme 2000; Kirovski et al. 2001; Lin et al. 2005; Seki and Kameyama 2003).

Related works

Currently, previous DRM solutions usually use four technologies in its process such as encryption, digital signature, digital watermarking, and fingerprinting (Hsu and Hou 2005; Lu and Liao 2003; Lu et al. 2006b; Tzeng et al. 2005; Wang and Chen 2007; Wang and Lin 2004; Zeng and Liu 1999).

In our understanding, the conventional DRM system (server-side encryption and user-side fingerprint embedding) was first proposed in Macq and Quisquater (1995). Then, it was extended by Bloom (2003) and Hartung and Girod (1997). In the method of Macq and Quisquater (1995), one global key-based encryption is needed to encode the original content at the server side. The encoded content can be sent to many users via network by multicasting. At the user side, the encoded content can be decrypted according to the global key. After that, a watermarked software (DRM controller software) is necessary for joint multimedia decryption and fingerprint embedding according to user's information. However, the watermarked software is still an open problem because *the original content is possibly revealed inside the system by this software* (*the original content leakage problem*) Lin et al. (2012). Therefore, users can save original contents without watermark information and distribute it via network.

In order to solve the original content leakage problem, Karthik and Hatzinakos (2007) proposed a joint fingerprinting and decryption (JFD) method. JFD employs the un-decrypted parts imitate multimedia fingerprint embedding in the decoding process. By using JFD, original content is not disclosed temporarily inside a system while decoding the content. However, the un-decrypted parts are the cause of distortion of digital content.

With the similar motivation, Chameleon method was proposed by Anderson and Manifavas (1997) based on secret table look up operations. In this scenario, the fingerprinted contents are decrypted for different users using different secure tables. Therefore, the Chameleon method can distinguish different users by checking the fingerprint for each secure table. However, the Chameleon method may consume greater bandwidth because each user needs a different secure table as the mention in Lian (2008).

With the different idea, Lin et al. (2012) proposed the *fingerprinting* method and user side using *vector quantization* domain (FVQ). FVQ employed the permutation and codeword substitution tables using static key-trees or dynamic key-trees. It seems it can save significant bandwidth and conveniently update key-trees. However, in the FVQ scheme, there is not trial content for users to try it before deciding whether to purchase or not.

To solve the original content leakage problem and to provide the trial content for users via network, our previous work (Thanh and Iwakiri 2014) employs the Huffman feature to implement the DRM system on a JPEG image. However, the amount of the embedding information is needed to improve in real applications.

Our contributions

In this paper, we describe a design and implementation of DRM technique based on an incomplete cryptography system. The know-how of the proposed method is the fundamental incomplete cryptography. Our method will degrade the quality of original contents to make the trial contents for delivering 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. We also join the watermarking process and decoding process to improve the problem of traditional DRM system of the original content leakage. Based on our method, we can control the quality of the decoded content according to the watermarked key.

In this study, the robustness of watermarking method is not considered. We only concentrate to solve the problem of a conventional DRM system which is to completely disclose the original content inside a user's system while decoding process. We combine two processes (decoding and watermarking) at the user side to become the incomplete decoding. The user's information is embedded into the decoded content simultaneously. Assuming that there are not any attacks on the decoded content, we always identify exactly the legal user by extracting the *userID* from the decoded content. From this idea, we make the following contributions in this paper:

1. We propose 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 present a new fragile fingerprinting method that includes trial contents for advertisement and *userID* for distinguishing the legal user. Our system makes it easier for users to try the digital content before purchasing.
3. Our proposed method can detect the source of pirated content by comparing the extracted *userID* from the incomplete decoded content with producer's database. It is considered that it can limit the illegal redistribution in advance.

Roadmap

This paper is organized as follows. The scheme of the proposed incomplete cryptography system is presented in the "Overview of incomplete cryptography" section. The implementation of digital content distribution system based on incomplete cryptography is explained in the "Implementation of digital content distribution system" section. The "Methods" section presents the algorithm of incomplete cryptography on the Joint Photographic Experts Group (JPEG) image. The experimental results with JPEG images are given in the "Results and discussion" section and the conclusion is summarized in the "Conclusions" section.

Overview of incomplete cryptography

The proposed incomplete cryptography for DRM systems is explained in this section. There are two steps in the proposed cryptography: the incomplete encoding and the incomplete decoding. The basic idea of the incomplete cryptography is shown in Fig. 1.

Incomplete encoding

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 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 distribute C widely to users as trial content via network.

Incomplete decoding

The incomplete decoding process is different from the complete decoding process. Decoded content is created by another decryption function D' with another decoded key $k'_i (i = 1, 2, ..., n)$. Note that, D' with k'_i is different from D with k, where D is the decoded function that can decode C to obtain P. 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 different from original content P. Therefore, the relationship of P and P'_i is $P'_i \neq P$ in incomplete cryptography system. Hence, this decoder process is quite different from the complete cryptography. This feature is called *incomplete decode*.

According to features of incomplete cryptography, if a set of the decoded keys k'_i with decoder function D' to decode a encoded C are chosen, a set of decoder contents P'_i will be created and different from each other. So, if 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.

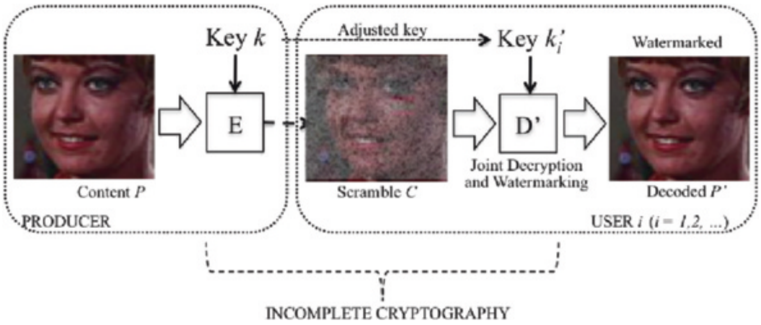


Fig. 1 Overview basic idea of incomplete cryptography

Implementation of digital content distribution system

We use the features of the incomplete cryptography for construction of the digital content distribution system. The incomplete cryptography is used to create the scrambled content (as trial content). The fragile watermarking is performed by using the incomplete cryptography.

Scrambled content

If the incomplete confidentiality characteristic of the incomplete cryptography is used, it is possible to make the scrambled content. The scrambled contents are used as trial contents, which are delivered to users via network. The scrambled contents has an important role in a user's decision of purchasing.

The basic idea of the scrambled algorithm is shown in Fig. 2(a). Suppose P is a content in the domain of original contents, k is an encoder key, and E is an encoder function. E will encode a part of P and degrade the quality of P. For instance, if a producer wants to make a scrambled picture C from P, he or she can use E to encode lower bits plane of pixels. The method of scrambled process is to use simultaneously for all pixels, and each pixel takes a different value. Therefore, the quality of P is degraded.

In the incomplete cryptography, an encoded part can be selected adaptively from many elements in digital content, so that a producer can prepare various scrambled contents to distribute via network as the trial contents. After selecting the part to encode, the formula (1) is used to create the scrambled content.

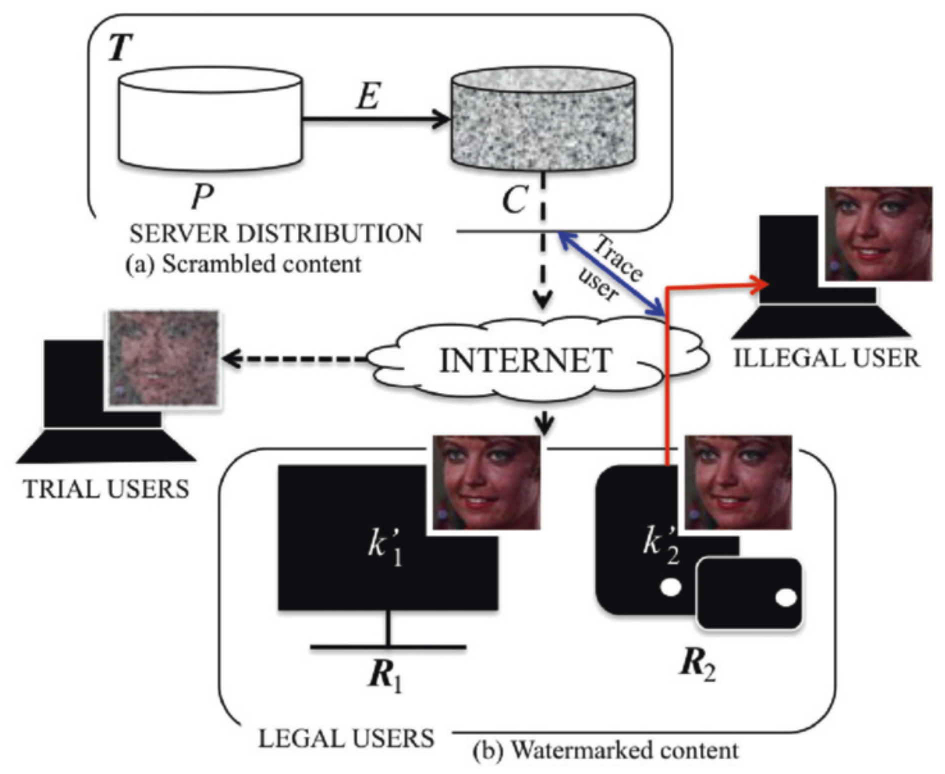


Fig. 2 DRM system based on incomplete cryptography (*a*, *b*)

Generation of watermarked key

In order to generate the watermarked key, T needs the user information that is registered by user R_i. The user information w_m may be userID, user's name, birthday, and so on. Using the key generation function G, T can create the watermarked key k'_i based on k, P and w_m,

$$k'_i = G(k, P, w_m) \tag{3}$$

In our work, we suppose that the watermarked key is delivered to users by safe way such as CD and USB. Therefore, the watermarked key is individually sent to a user and it is not tampered by any attackers.

Watermarked content

In the proposed method, the decoded content P'_i is different from the original content P. Assume that a user R_i can decode C to obtain P'_i that closes to the quality of P, then we can propose a watermark algorithm to control quality of P'_i. The watermarked algorithm is explained as follows (see Fig. 2(b)):

Suppose a user R_i receives a decoder key k'_i from T and decodes scrambled C. Here, if $k'_i \neq k$, it is clear that $P'_i \neq P$. However, as shown in Fig. 2(b), if the quality of P'_i is sufficient for the user, even if $k'_i \neq k$, user cannot notice the distortion of P after the watermark embedding is applied.

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 using k'_i to make P'_i, P'_i is not only decoded with slight deterioration, but also watermarked with particular information (i.e., user information). It is the elemental mechanism of fragile watermarking based on the incomplete cryptography system.

Assuming that one legal user redistributed the watermarked content to illegal user, it is difficult to trace the redistributed source without user's information. 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 database, the user is a legal user. Conversely, if the watermarking information is different from his 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 the producer about the existence of watermarking which can exactly identify users, and to limit the illegal redistribution in advance.

Methods

In this section, an algorithm which is applied to JPEG image (International Telecommunication Union 1992) is explained. The scrambled image and the watermarked image are generated based on the incomplete cryptography.

Summary of JPEG algorithm

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 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 often become "0" after quantization, the spectrum of picture tends to be constructed with low-frequency coefficients.

Our proposed methods

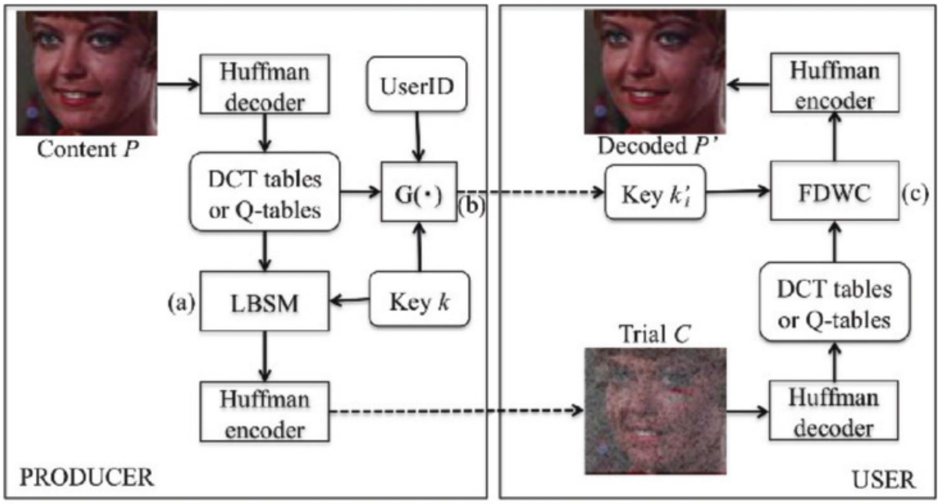
To make the scrambled contents and the incomplete decoding contents of JPEG, we have selected the quantized DCT coefficients to implement the incomplete cryptography. There are two reasons for choosing of the DCT coefficients. The first one is that it is easy to control the quality of the JPEG image. The second one is that it is flexible by selecting a luminance component (Y component) or two chrominance components (UV component) in the quantized DCT coefficients in order to implement our proposed method.

Additionally, in order to compare the efficiency of the above components, we alter the DCT quantization table to control the quality of the JPEG image based on the Y coefficient and UV coefficient. There are two types of a DCT quantization table such as the table for the UV component and the table for Y component. In each quantization table, there are 64 quantized coefficients. If we alter these quantization tables, we can easily control the quality of digital content with low computational cost.

Figure 3 describes the overview of our proposed system. First, the Huffman decoder is performed and the DCT tables or DCT quantization tables (Q-tables) are obtained. Then, we apply the proposed lower bits scramble method (LBSM) to create the scrambled content (trial content). The decoding key of an individual user is generated based on the function G by using *userID*. In order to decode the trial content, the Huffman decoder is performed and then the DCT tables and Q-tables are decoded by the proposed fragile decryption and watermarking combination (FDWC). In this process, the *userID* is also embedded into the decoded content. The details of LBSM and FDWC are described as follows.

Our lower bits scramble method

Let us denote the (i,j)th DCT coefficient of the lth table as $S_l(i,j), 0 \leq i,j \leq 8, l = 1, ..., B$, where B is the total number of tables in the image P. Suppose that, the specified coefficient $S_l(i,j)$ is selected from the quantized DCT coefficients. We proposed a new LBSM to control the least significant bits of the DCT coefficient for making the scrambled content (see Fig. 3(a)). In this method, the lower bits, except for the most significant bit (MSB)



※ Q-tables: DCT quantization table
※ G(•): Key generation function

Fig. 3 Overview of the system (*a*–*c*)

of “1” in $S_l(i,j)$, are encoded by the random key to scramble JPEG image. The scrambled method is described as follows:

Step 1. The encoding key k consists the sequence of $k_l(i,j)$ that is generated to scramble a quantized DCT coefficient $S_l(i,j)$.
Step 2. A shift coefficient m for each $S_l(i,j)$ can be obtained by the following calculation:

$$m \leftarrow 8 - \lfloor \log_2(|S_l(i,j)|) \rfloor, \tag{4}$$

where the symbol $\lfloor . \rfloor$ is the floor function meaning “the greatest integer less than or equal to”. $k_l(i,j)$ is shifted m bits to prepare the encryption key $k_l^m(i,j)$.

$$k_l^m(i,j) \leftarrow k_l(i,j) \gg m. \tag{5}$$

Step 3. The significant bits of $S_l(i,j)$ are encoded with using $k_l^m(i,j)$ to make scrambled content C. The scrambled coefficient of C is given by,

$$S'_l(i,j) \leftarrow S_l(i,j) \oplus k_l^m(i,j), \tag{6}$$

where the symbol $\oplus$ is the XOR function.

Table 1 shows the positions of the encoded bits in this process. In this table, “$*$” denotes the original bit of quantized DCT coefficient, and “e” denotes the encrypted position bits.

Table 1 Scramble the DCT coefficient

DCT coefficient $S_l(i,j)$	$\lfloor log_2 S_l(i,j) \rfloor$	Scrambled $S'_l(i,j)$
1*******	7	1eeeeeee
01******	6	01eeeeee
001*****	5	001eeeee
0001****	4	0001eeee
00001***	3	00001eee
000001**	2	000001ee
0000001*	1	0000001e

Thus, $S_l(i,j)$ is substituted by another value, and the quality of P is partially maintained to make the scrambled content C.

In the scrambled method, the scrambled content C is disclosed as a part of P because the MSB of $S_l(i,j)$ is not encoded . This method uses the *incomplete encoding* feature of the incomplete cryptography. C is widely distributed to users as a trial content via network.

Generation of fragile watermarking key

In order to prepare the decoded key for each user, as shown in Fig. 3(b), T generates a decryption key k'_i by using a $UserID : W = \{w_t, 1 \leq t \leq M \times M\}$, then sends it to R_i. Here, the *incomplete decoding* feature of incomplete cryptography is used. The generation of decoded key is described as follows:

Step 1. T extracts w_t from *userID* and embeds it into the least significant bit (LSB) of key k'_i consisting sequence of $k'_l(i,j)$ as follows.

$$k'_l(i,j) \leftarrow \begin{cases} k^m_l(i,j) \oplus (S_l(i,j) \,\&\, 0 \times 01) \ \ (\text{if } w_t = 0), \\ k^m_l(i,j) \oplus (\overline{S_l(i,j)} \,\&\, 0 \times 01) \ \ (\text{if } w_t = 1), \end{cases} \tag{7}$$

where the symbol "&" is the AND function and the symbol $\overline{S_l(i,j)}$ is NOT function of $S_l(i,j)$.

Step 2. T prepares watermarked key k'_i based on **Step 1**, and delivers to R_i.

Here, we suppose that the decoded key k'_i is safely delivered to R_i by CD, USB, etc.

Our fragile decryption and watermarking combination

After receiving the decoded key, R_i can decode the scrambled content by using the decoded key k'_i. In this process, the scrambled DCT coefficients $S'_l(i,j)$ in C are decoded to close original coefficient with the watermark information that is embedded while the decoding process (see Fig. 3(c)). We call this process fragile decryption and watermarking combination (FDWC). The detail of FDWC is explained as follows.

Step 1. R_i extracts the scrambled DCT coefficient $S'_l(i,j)$ from C.

Step 2. R_i extracts the corresponding element $k'_l(i,j)$ of the decoded key k'_i.

Step 3. Each scrambled DCT coefficient of C is decoded as follows:

$$S''_l(i,j) \leftarrow S'_l(i,j) \oplus k'_l(i,j). \tag{8}$$

In FDWC, the user information (*userID*) is embedded into the LSB of the quantized DCT coefficient $S''_l(i,j)$ at some particular position of DCT tables.

Table 2 shows the decoded positions bit in this process. The watermarking information (*userID*) is embedded at the position of "w". This FDWC is the most basic decoded

Table 2 Incomplete decode the DCT coefficient

Scrambled $S'_l(i,j)$	$\lfloor log_2 S'_l(i,j) \rfloor$	Watermarked $S''_l(i,j)$
`1eeeeeee`	7	`1******w`
`01eeeeee`	6	`01*****w`
`001eeeee`	5	`001****w`
`0001eeee`	4	`0001***w`
`00001eee`	3	`00001**w`
`000001ee`	2	`000001*w`
`0000001e`	1	`0000001w`

processing in the incomplete cryptography. The advantage of incomplete cryptography is that when decoding the scrambled content C, watermarking information is simultaneously embedded into the decoding content. Therefore, it is possible to implement the watermarked process while decoding.

Additionally, when a producer verifies the legal user of content P, he/she can extract the user information from the LSB of a particular DCT coefficient by using the secret key k_s. In this paper, k_s is the LSB of each DCT coefficient in P'.

According to this proposed method, a producer is possible to verify the legal user of digital content, so that, he/she can easily manage the copyright of digital contents.

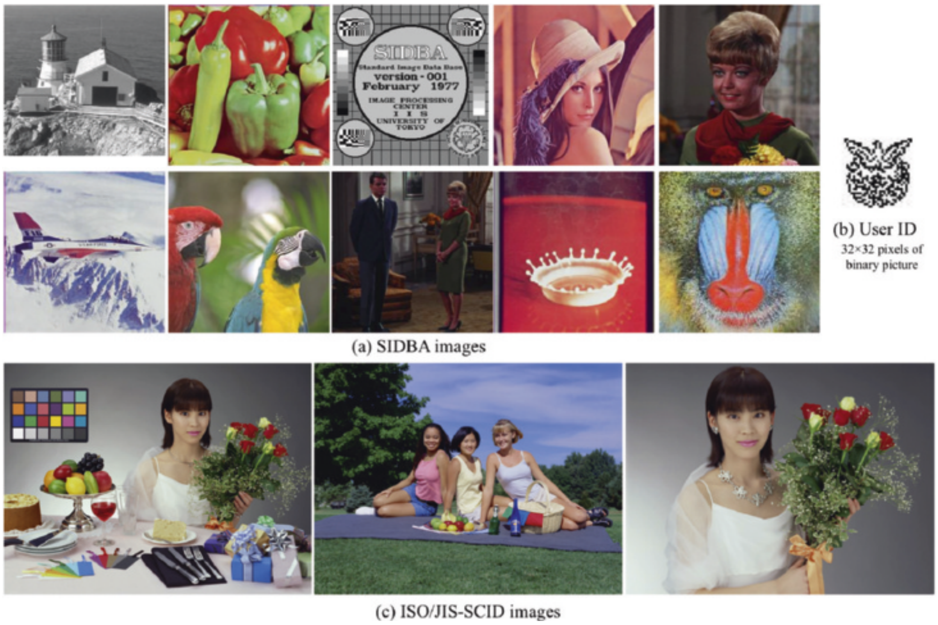
Results and discussion

Experimental environment

All experiments are performed by incomplete encoding and incomplete decoding on the JPEG image using the Vine Linux 3.2 system. In order to generate the encryption key 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 to view the experimental JPEG images.

Experimental image

We prepare some different features of the experimental images regarding CG, scenery, construction, and person. Ten test images are the 8-bit RGB images of Standard Image Data BAse (SIDBA) international standard image (Lighthouse, Pepper, Title, Lenna, Girl, Airplane, Parrots, Couple, Milkdrop, Mandrill) with size 256 × 256 pixels (Fig. 4(a)). We use the additional database images ISO/JIS-SCID (Party, Picnic, Portrait) with size 2048 × 1536 pixels, 8-bit RGB (Fig. 4(c)). Here, all images are compressed with quality 75 (the lowest 0 ↔ 100 the highest) to make experimental JPEG images for evaluation of the proposal method.



(a) SIDBA images

(b) User ID 32×32 pixels of binary picture

(c) ISO/JIS-SCID images

Fig. 4 The experimental images (*a*–*c*)

We prepare a bitstream $M \times M = 32 \times 32$ pixels of the binary picture (UserID) as watermarking information (see Fig. 4(b)).

Evaluation of image quality

We use peak signal to noise ratio (PSNR) (Matsui 1998) 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

$$\mathrm{PSNR} = 20 \log \frac{255}{MSE} \quad [\mathrm{dB}] \tag{9}$$

$$\mathrm{MSE} = \sqrt{\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \{g(i,j) - g'(i,j)\}^2}$$

(MSE: Mean Square Error).

In these experiments, the PSNR values are 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 (Matsui 1998). First, we evaluated the relationship of the subjective image quality and PSNR. Here, we prepared 10 images with quality of 15–32 dB in PSNR. Those images were controlled with DCT coefficients of Y and UV component, respectively. After that, the experimental JPEG images are assessed subjectively with 10 testers and the mean opinion score (MOS) grade is calculated.

The MOS is an arithmetic mean of all individual scores by tester, and can range from 1 grade (worst) to 5 grade (best). In the experiment, MOS was also reported is perceived quality of test JPEG images. The MOS values are assigned based on the values shown in Table 3.

In the MOS experiment, Lenna, Lighthouse, Pepper, and Title are used as test images.

In our MOS test, the test JPEG images are randomly shown to testers. The MOS grades of each image are decided by each tester.

Figure 5 shows the relation between MOS and PSNR. According to Fig. 5, we realize that the testers feel the deterioration when PSNR of image is lower than approximately 22 dB (MOS: 0–2.5). In addition, when PSNR is between 22 and 29 dB (MOS: 2.5–3.5), the testers feel 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 conclude that the PSNR of the scrambled content is appropriately between 22 and 29 dB, and the PSNR of the incomplete decoding should be higher than 29 dB.

Results and analysis

This section presents some empirical results concerning of LBSM, FDWC on the JPEG images. First, the quantized DCT coefficients (position: $(i,j), (0 \leq i \leq 7, 0 \leq j \leq 7)$)

Table 3 MOS grading evaluation

MOS grade	Description
5	Deterioration is imperceptible
4	Deterioration is perceptible but not annoying
3	Degradation is slightly annoying
2	Deterioration is annoying
1	Deterioration is very annoying

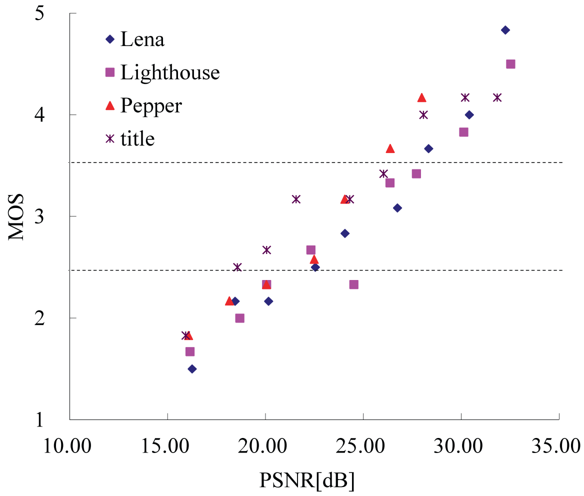


Fig. 5 The correspondence of MOS and PSNR

are extracted from the DCT table. Then, the proposed method is applied to a DCT coefficient (0,0) of a Y component (YDCT), a DCT coefficient (0,0) of a UV component (UVDCT), the low frequency coefficients excluding a DCT coefficient (0,0) of a Y component (YADC), low frequency coefficient excluding a DCT coefficient (0,0) of Y and UV components (ADCT), respectively. The details of these experimental methods are shown in Fig. 6. Here, the DCT coefficient (0,0) is the most low-frequency coefficient. To restrict the experimental zone of the DCT coefficient, we use the condition $i + j \leq \alpha$ (where α is the zone restriction factor). In this paper, we decide to implement DCT coefficients with $\alpha = 3$.

Besides, the DCT quantization tables are also extracted, and its quantized coefficients are also applied using the proposed method. We recognize that there are many non-effective quantized coefficients in the Q-table for the high-frequency component. It is

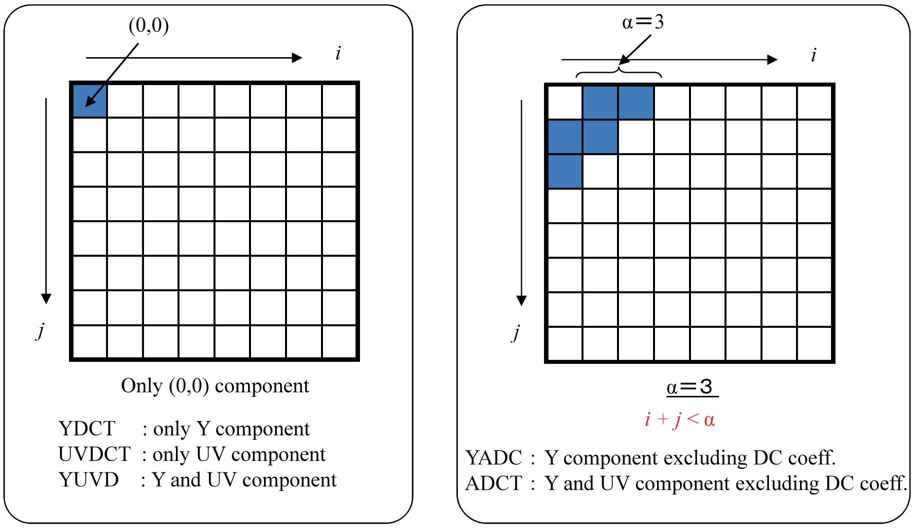
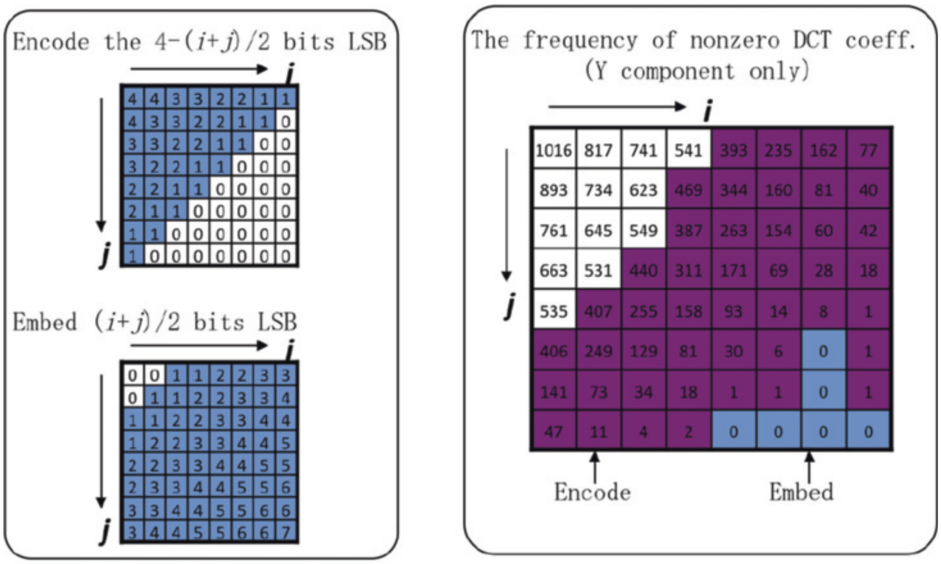

(0,0)
i
j
Only (0,0) component
YDCT : only Y component
UVDCT : only UV component
YUVD : Y and UV component
α=3
i
j
α=3
i + j < α
YADC : Y component excluding DC coeff.
ADCT : Y and UV component excluding DC coeff.

Fig. 6 Experimental DCT domain method

desirable to target for LBSM and FDWC. Therefore, we decide to encode $4 - (i + j)/2$ LSB bits (minimum: 0 bit) of quantized coefficients in Q-table and embed $(i + j)/2$ information bits (maximum: 8 bits) into each quantized coefficients (WQTB). The details of these experimental methods are shown in Fig. 7(a). On the other hand, we calculate the number of non-zero DCT coefficients in the whole JPEG image. We encode the quantized coefficients that has the number of non-zero DCT coefficient below 500, and embed information into the quantized coefficients that has no non-zero DCT coefficient frequency (AQTB: Fig. 7(b)). Since the AQTB method embeds the information into the quantized coefficients that has no non-zero DCT coefficient frequency, the decoded image (embedded image) is not degraded in the decoding process and all bits of such kind of quantized coefficients can be substituted by information bits.

Here, we show an example for processing of incomplete cryptography. In order to make a scrambled content, the quantized DCT coefficient $S_l(i,j)$ is extracted from the DCT table, the encryption key k is generated to scramble the lower bits except the MSB of bit "1" in the $S_l(i,j)$. Then, scrambled content C is created by incomplete encoding. When decoding C, the LSB of $S'_l(i,j)$ is substituted by watermarking information (user individual). Assume that $S_l(i,j) = 21$; then $S_l(i,j)$ can be expressed as binary bits $S_l(i,j) = 10101_2$. As in the scrambled method, we generate an encoded key $k_l(i,j)$, and set that as $k_l(i,j) = 31(k_l(i,j) = 11111_2)$ for example. For encoding $S_l(i,j)$, we calculate $m = 8 - \lfloor \log_2(|21|) \rfloor = 4$ (see formula (4)). m is used to create $k_l^m(i,j) = k_l(i,j) \gg m = 31 \gg 4$; then, $k_l^m(i,j) = 00001_2$. Finally, following (6), $S'_l(i,j) = S_l(i,j) \oplus k_l^m(i,j) = 10101_2 \oplus 00001_2 = 10100_2 (S'_l(i,j) = 20)$ is scrambled. To illustrate the decoded process, let us assume the watermark bit as $w = 1$, then $k'_l(i,j) = k_l^m(i,j) \oplus (\overline{S_l(i,j)} \ \& \ 0 \times 01) = 00001_2 \oplus (\overline{10101_2} \ \& \ 0 \times 01) = 00001_2$ (see formula (7)). If $k'_l(i,j)$ is used to decode $S'_l(i,j)$, we can obtain $S''_l(i,j) = S'_l(i,j) \oplus k'_l(i,j) = 10100_2 \oplus 00001_2 = 1010\underline{\mathbf{1}}_2$. It means that the watermark bit $w = 1$ is embedded into LSB of $S''_l(i,j)$. To extract the watermark from the


Encode the 4-(i+j)/2 bits LSB
i
j

4	4	3	3	2	2	1	1
4	3	3	2	2	1	1	0
3	3	2	2	1	1	0	0
3	2	2	1	1	0	0	0
2	2	1	1	0	0	0	0
2	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

Embed (i+j)/2 bits LSB
i
j

0	0	1	1	2	2	3	3
0	1	1	2	2	3	3	4
1	1	2	2	3	3	4	4
1	2	2	3	3	4	4	5
2	2	3	3	4	4	5	5
2	3	3	4	4	5	5	6
3	3	4	4	5	5	6	6
3	4	4	5	5	6	6	7

The frequency of nonzero DCT coeff.
(Y component only)
i
j

1016	817	741	541	393	235	162	77
893	734	623	469	344	160	81	40
761	645	549	387	263	154	60	42
663	531	440	311	171	69	28	18
535	407	255	158	93	14	8	1
406	249	129	81	30	6	0	1
141	73	34	18	1	1	0	1
47	11	4	2	0	0	0	0

(a) WQTB : Y and UV component

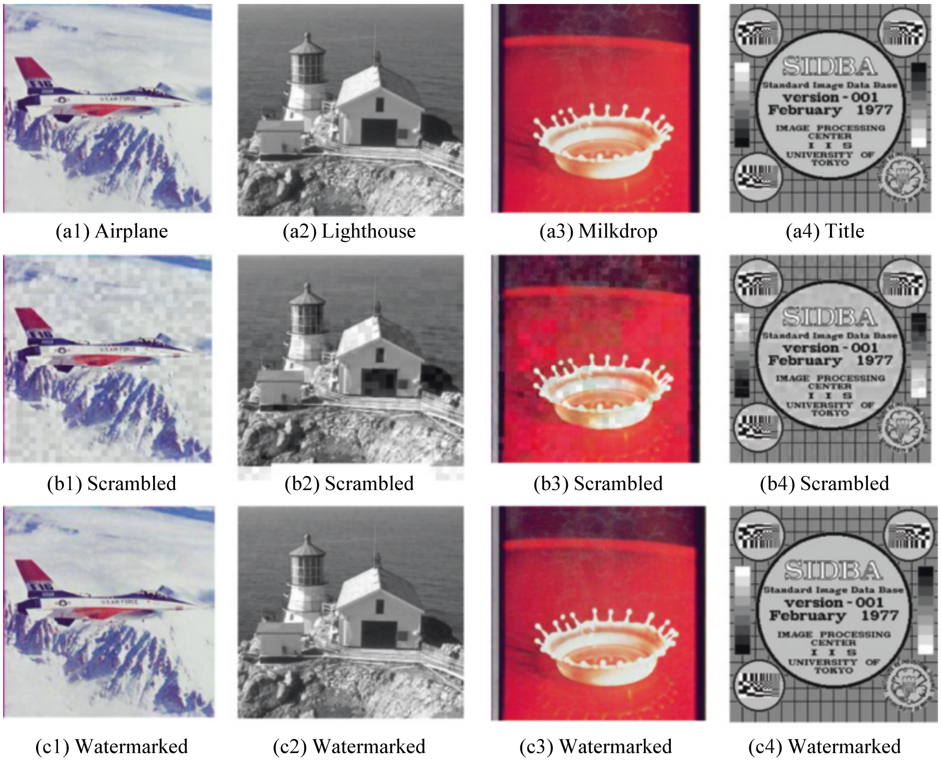
(b) AQTB : Y and UV component

Fig. 7 a, **b** Experimental Q-table method watermarked JPEG image, it can be extracted from LSB (as the extract key k_s) of $S''_l(i,j)$ and compare with *userID* to confirm the illegal user.

The experimental results are shown in Fig. 8 and Table 4. We see that the watermarked JPEG images are not distinguishable from the original JPEG images. The scrambled JPEG images are degraded about 20 dB, and they seem appropriate as a trial content. We calculate PSNR values of the output JPEG images in every processes and extract the watermark information (embedded binary data) perfectly from the incomplete decode JPEG images. In our method, we extract the LSB of quantized DCT coefficients.

Figure 9 is an experimental sample of Girl images. According to the results in Fig. 9, it is possible to produce the scrambled content (see Fig. 9(c)) and incomplete decoded content (see Fig. 9(d)) based on the incomplete cryptography. Furthermore, the watermark can be extract accurately (see Fig. 9(e)). We also compared the results of YDCT and UVDCT as in Table 4, and confirm that when we adjust the UV component, the image deterioration was extremely more conspicuous than that applied to the Y component. Therefore, we can make the scrambled content efficiently with drawing up on the least UV component. However, because the image deterioration is not conspicuous when implementing the Y component, it is better to embed the abundant watermark information into the decoded content for keeping the quality of the decoded image.

Table 4 shows the experimental results using the large size JPEG images (ISO/JIS-SCID). The scrambled image and watermarked image are created by the proposed method. If the size of a JPEG image is large, a large amount of watermarking information is embedded in the image.


(a1) Airplane (a2) Lighthouse (a3) Milkdrop (a4) Title

(b1) Scrambled (b2) Scrambled (b3) Scrambled (b4) Scrambled

(c1) Watermarked (c2) Watermarked (c3) Watermarked (c4) Watermarked

Fig. 8 Experimental results of ADCT method (*a1–c4*)

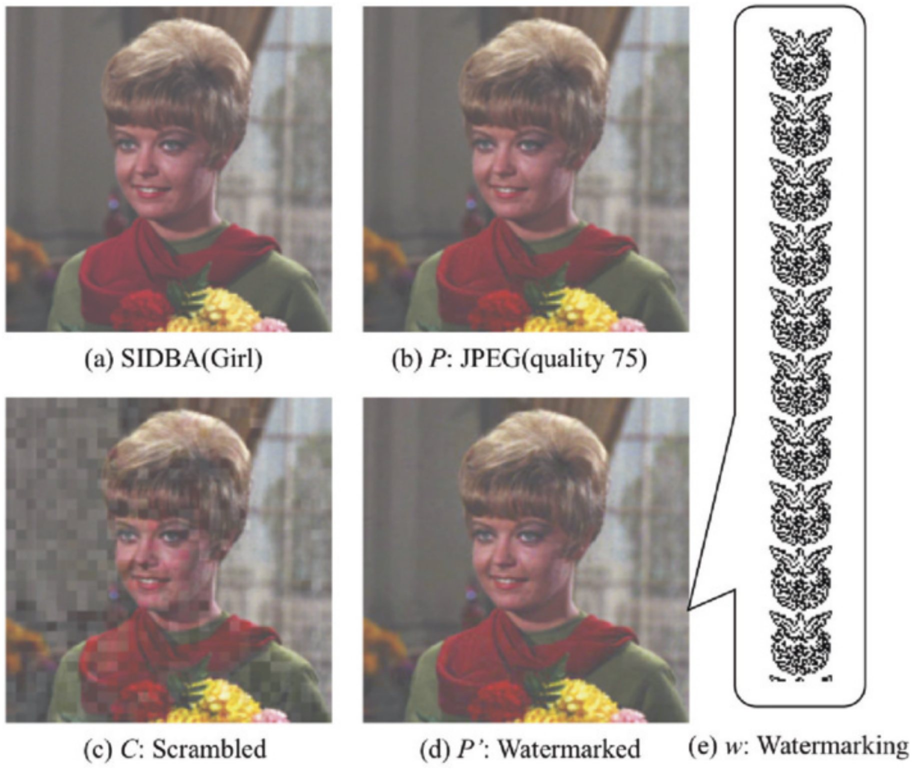
Table 4 PSNR [dB] and embedded bits (DCT coefficient)

Method	Image	*P*	*C*	P'	Emb. [bit]	α
YDCT	Title	31.84	27.80	31.75	1024	0
	Lenna	32.37	26.93	32.32	1024	0
	Girl	32.71	26.65	32.63	1024	0
	Airplane	30.20	25.16	30.17	1024	0
	Parrots	34.25	27.98	34.16	1024	0
	Couple	34.06	25.77	33.96	1024	0
	Milkdrop	31.94	27.36	31.88	1024	0
	Mandrill	24.96	23.57	24.95	1024	0
	Lighthouse	32.67	27.59	32.59	1024	0
	Pepper	28.81	25.69	28.78	1024	0
	Party	35.16	27.85	35.05	*49152*	0
	Picnic	34.39	27.61	34.28	*49152*	0
	Portrait	35.99	27.58	35.86	*49152*	0
UVDCT	Title	31.84	26.72	31.71	512	0
	Lenna	32.37	27.48	32.65	512	0
	Girl	32.71	24.40	32.57	512	0
	Airplane	30.19	24.42	30.12	512	0
	Parrots	34.25	24.16	34.03	512	0
	Couple	34.06	19.07	33.85	512	0
	Milkdrop	31.94	27.51	31.81	512	0
	Mandrill	24.96	21.03	24.93	512	0
	Lighthouse	32.67	26.68	32.62	512	0
	Pepper	28.81	24.78	28.73	512	0
	Party	35.16	16.87	34.91	*24576*	0
	Picnic	34.39	20.39	34.17	*24576*	0
	Portrait	35.99	15.47	35.69	*24576*	0
YUVD	Title	31.84	26.80	31.74	1536	0
	Lenna	32.37	24.11	32.19	1536	0
	Girl	32.71	23.01	32.49	1536	0
	Airplane	30.20	22.19	30.09	1536	0
	Parrots	34.25	22.93	33.92	1536	0
	Couple	34.06	17.72	33.75	1536	0
	Milkdrop	31.94	24.65	31.75	1536	0
	Mandrill	24.96	20.35	24.92	1536	0
	Lighthouse	32.67	27.58	32.60	1536	0
	Pepper	28.81	23.75	28.71	1536	0
	Party	35.16	16.56	34.82	*73728*	0
	Picnic	34.39	19.19	34.05	*73728*	0
	Portrait	35.99	15.17	35.	*73728*	0
YADC	Title	31.84	28.33	31.60	4592	3
	Lenna	32.37	28.52	32.19	5156	3
	Girl	32.71	26.94	32.49	4975	3
	Airplane	30.20	26.09	30.11	4973	3
	Parrots	34.25	29.74	33.93	4872	3
	Couple	34.06	25.69	33.76	4756	3
	Milkdrop	31.94	27.69	31.72	4819	3
	Mandrill	24.96	24.40	24.92	5738	3
	Lighthouse	32.67	27.99	32.44	4935	3
	Pepper	28.81	26.70	28.72	5397	3
	Party	35.16	28.32	34.83	*172976*	3
	Picnic	34.39	27.94	34.08	*211230*	3
	Portrait	35.99	28.19	35.67	*183472*	3

Table 4 PSNR [dB] and embedded bits (DCT coefficient) *(Continued)*

ADCT	Title	31.84	28.33	31.60	4592	3
	Lenna	32.37	27.03	31.54	7015	3
	Girl	32.71	26.79	31.88	6507	3
	Airplane	30.20	25.99	29.77	6404	3
	Parrots	34.25	28.26	32.96	6702	3
	Couple	34.06	25.65	33.11	6082	3
	Milkdrop	31.94	25.29	31.14	6543	3
	Mandrill	24.96	23.96	24.76	7920	3
	Lighthouse	32.67	27.99	32.44	4935	3
	Pepper	28.81	25.26	28.34	7716	3
	Party	35.16	27.97	34.18	*220517*	3
	Picnic	34.39	27.35	33.33	*273189*	3
	Portrait	35.99	28.01	35.96	*183472*	3

Moreover, the experimental results of DCT Q-table are also shown in Fig. 10 and Table 5. From these results, we recognize that it is equivalent to DCT coefficient methods. It means that, it is possible to produce the scrambled content and incomplete decoding content. Especially, AQTB method can provide the non-degraded content which is embedded about 400 bits information. In other words, DCT quantization method can control the quality of content with low computation cost. On the other hand, DCT quantization table methods have a capacity lower than that of DCT coefficient methods.


(a) SIDBA(Girl)

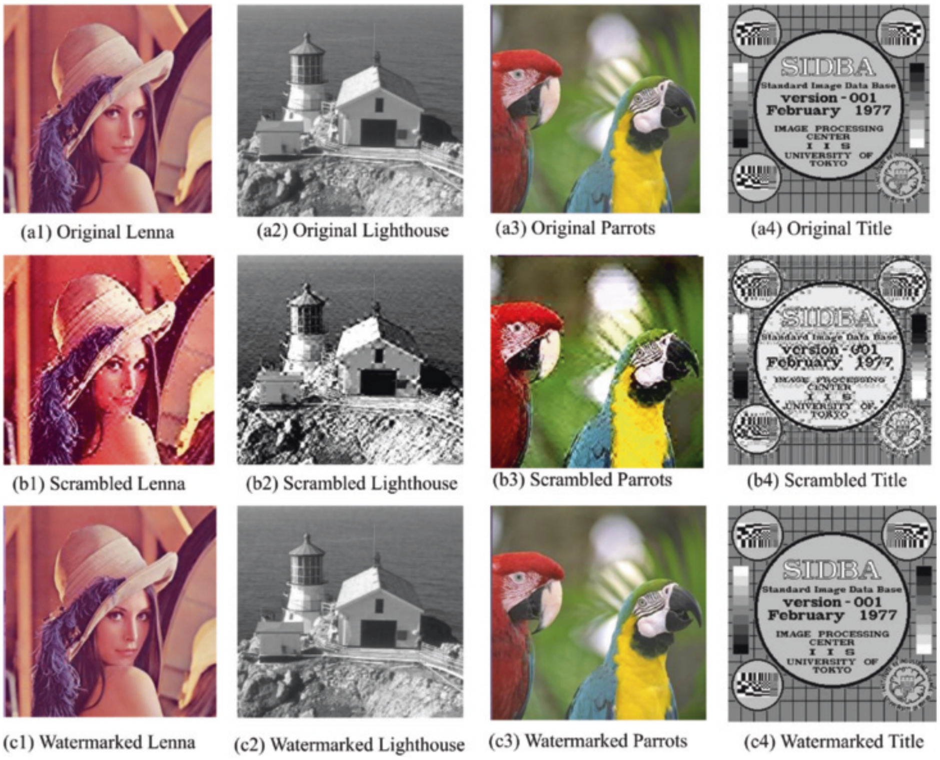
(b) *P*: JPEG(quality 75)

(c) *C*: Scrambled

(d) *P'*: Watermarked

(e) *w*: Watermarking

Fig. 9 **a**–**e** Examples of girl images (ADCT)



(a1) Original Lenna (a2) Original Lighthouse (a3) Original Parrots (a4) Original Title

(b1) Scrambled Lenna (b2) Scrambled Lighthouse (b3) Scrambled Parrots (b4) Scrambled Title

(c1) Watermarked Lenna (c2) Watermarked Lighthouse (c3) Watermarked Parrots (c4) Watermarked Title

Fig. 10 Experimental results of WQBT method (*a1*–*c4*)

According to the above results, we have established the incomplete cryptography system based on the proposed method. Scrambled content is created to disclose the original content and distributed widely to users by using LBSM. In FDWC, we change the quantized DCT coefficient itself instead of the LSB of quantized DCT coefficient by a devised decryption key. Thus, the original content is not decoded temporarily inside the system. Thus, we conclude that the above technical problem by the conventional DRM system is solved by using the incomplete cryptography system.

Comparison of our proposed method with related work

According to analytics of related works, JFD seems to be promising to achieve decryption and fingerprint embedding at the same time. However, since un-decrypted parts in JFD are employed as fingerprinting information for user, then the quality of the decrypted content is limited. Our method uses the *userID* that is embedded into the specified position; therefore, our method can flexibly control the quality of decrypted content. In addition, in our method, the incompletely decrypted blocks (watermarked blocks) are used instead of un-decrypted blocks; therefore, our method can take better trade-off between multimedia security and fingerprinting imperceptibility than JFD . The detail of comparison of our method with JFD is shown in Table 6.

Conclusions

In this paper, we have presented a scheme of an incomplete cryptography system and proposed the digital content distribution system based on incomplete cryptography. This

Table 5 PSNR [dB] and embedded bits (quantization table)

Method	Image	P	C	P'	Emb. [bit]	α
WQTB	Title	31.84	13.18	28.99	416	0
	Lenna	32.37	17.49	31.72	416	0
	Girl	32.71	17.98	32.48	416	0
	Airplane	30.20	14.95	29.66	416	0
	Parrots	34.25	17.34	33.64	416	0
	Couple	34.06	19.78	33.76	416	0
	Milkdrop	31.94	27.36	31.88	1024	0
	Mandrill	24.96	23.57	24.95	1024	0
	Lighthouse	32.67	16.45	31.52	416	0
	Pepper	28.81	17.08	28.52	416	0
	Party	35.16	27.85	35.05	*49152*	0
	Picnic	34.39	27.61	34.28	*49152*	0
	Portrait	35.99	27.58	35.86	*49152*	0
AQTB	Title	31.84	15.52	31.84	512	0
	Lenna	32.37	14.85	32.37	416	0
	Girl	32.71	23.10	32.71	424	0
	Airplane	30.20	19.61	30.20	336	0
	Parrots	34.25	20.31	34.25	344	0
	Couple	34.06	23.54	34.06	464	0
	Milkdrop	31.94	27.51	31.81	512	0
	Mandrill	24.96	21.03	24.93	512	0
	Lighthouse	32.67	18.27	32.67	512	0
	Pepper	28.81	17.40	28.81	344	0
	Party	35.16	16.87	34.91	*24576*	0
	Picnic	34.39	20.39	34.17	*24576*	0
	Portrait	35.99	15.47	35.69	*24576*	0

approach integrates the encoding process and watermarking progress of DRM technology. By doing so, we can eliminate the problem of the present DRM technology and manage the legal user effectively.

One of the lessons learned from this paper is that in order to make the scrambled image and the incomplete decoded image for JPEG, it is possible to process the Y component and UV component flexibly. Also, another lesson is that we can control the incomplete decoded image quality using a specialized key individually. Subsequently, the watermark information is correctly extracted from the watermarked image by using this approach. The watermarked images are in good visual quality and have high PSNR values. The effectiveness of the proposed scheme has been demonstrated with the aid of experimental results.

Therefore, we conclude that the proposed method is useful for the rights management technology in illegal content distribution via network.

Table 6 Comparison between our proposed method with JFD

	Proposed method	JFD (Karthik and Hatzinakos 2007)
Domain	Partial encryption/decryption	Partial encryption/decryption
Block	Incompletely/completely decryption	Un-decrypted/decrypted
Coefficient	Watermarked	Un-decrypted/decrypted
Fingerprint	UserID	Un-decrypted part

Endnotes

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

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

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MI conceived of the idea for proposed method and participated in its design. TMT designed the idea, implemented the experiments on JPEG image, and drafted the manuscript. Both authors read and approved the submitted manuscript.

Acknowledgements

This work is supported by our lab members, who help us to evaluate the MOS experiments.

Author details

[1] Department of Computer Science, National Defense Academy, 1-10-20, Hashirimizu, Yokosuka-shi, Kanagawa 239-8686, Japan. [2] Department of Network Security, Le Quy Don Technical University, 236 Hoang Quoc Viet, Cau Giay, Hanoi, Vietnam. [3] Department of Computer Science, Tokyo Institute of Technology, 2-12-2, Ookayama, Meguro-ku, Tokyo 152-8552, Japan.

Received: 24 February 2015 Accepted: 7 June 2015
Published online: 08 July 2015

References

Anderson RJ, Manifavas C (1997) Chameleon, a new kind of stream cipher. In: Proceedings of the fourth international workshop on fast software encryption. LNCS, Springer Berlin Heidelberg Vol. 1267. pp 107–113

Bloom J (2003) Security and rights management in digital cinema. In: Proc. of the IEEE international conference on acoustics, speech and signal processing Vol. 4. pp 712–715. DOI: 10.1109/ICASSP.2003.1202742

Boneh D, Shaw J (1998) Collusion-secure fingerprinting for digital data. IEEE Trans Inform Theory 44:1897–1905

Cox IJ, Bloom JA, Miller ML (1999) Digital watermarking. Morgan Kaufmann Publishers, Burlington, Massachusetts

Emmanuel S, Kankanhalli MS (2003) A digital rights management scheme for broadcast video. Multimed Syst 8:444–458

Hartung F, Girod B (1997) Digital watermarking of MPEG-2 coded video in the bitstream domain. In: Proc. of the IEEE international conference on acoustics, speech and signal processing Vol. 4. pp 2621–2624. DOI: 10.1109/ICASSP.1997.595326

Hartung F, Ramme F (2000) Digital rights management and watermarking of multimedia content for m-commerce applications. In: IEEE Communications Magazine, Selected Papers from ISS2000 Vol. 38. pp 77–84. DOI: 10.1109/35.883493

Hsu CS, Hou YC (2005) Copyright protection scheme for digital images using visual cryptography and sampling methods. Opt Eng 44(7):1–10

International Telecommunication Union (1992) The International Telegraph and Telephone Consultative Committee Information Technology - Digital Compression and Coding of Continuous-tone still Images - Requirements and Guidelines

Karthik K, Hatzinakos D (2007) Decryption key design for joint fingerprinting and decryption in the sign bit plane for multicast content protection. I J Network Secur 4(3):254–265

Katzenbeisser S, Petitcolas FAP (2000) Information hiding technique for steganography and digital watermarking. Artech House, London, United Kingdom

Kirovski D, Peinado M, Petitcolas FAP (2001) Digital rights management for digital cinema. In: International Symposium on Optical Science and Technology, Security in Imaging Vol. 4472, XXIV, 105, DOI: 10.1117/12.449745

Lin ET, Eskicioglu AM, Lagendijk RL, Delp EJ (2005) Advances in digital video content protection. Proc IEEE 93(1):171–183

Lin C-Y, Prangjarote P, Kang L-W, Huang W-L, Chen T-H (2012) Joint fingerprinting and decryption with noise-resistant for vector quantization images. Signal Process 92(9):2159–2171

Lian S (2008) Multimedia content encryption: techniques and applications. CRC Press (Auerbach Publications)

Lu CS, Liao HYM (2003) Structural digital signature for image authentication – an incidental distortion resistant scheme. IEEE Trans Multimed 5(2):161–173

Lu CS, Sun SW, Hsu CY, Chang PC (2006a) Media Hash-dependent image watermarking resilient against both geometric attacks and estimation attacks based on false positive-oriented detection. IEEE Trans Multimed 8(4):668–685

Lu ZM, Zheng WM, Pan JS, Sun Z (2006b) Multipurpose image watermarking method based on mean-removed vector quantization. J Inf Assur Secur 1:33–42

Matsui K (1998) Fundamentals of digital watermarking. Morikita-publisher (in Japanese). 1-4-11, Fujimi, Chiyoda-ku, Tokyo 102-0071 Japan. (in Japanese)

Macq BM, Quisquater JJ (1995) Cryptology for digital TV broadcasting. Proc IEEE 83(6):944–957

Seki A, Kameyama W (2003) A proposal on open DRM system coping with both benefits of rights-holders and users. In: IEEE conference on image proceedings Vol. 7. pp 4111–4115. DOI: 10.1109/GLOCOM.2003.1259001

Shi C, Bhargava B (1998) A fast MPEG video encryption algorithm. In: Proceedings of the ACM International Conference on Multimedia. pp 81–88. ISBN: 0-201-30990-4, ACM New York, NY, USA

Sun SW, Chen JR, Lu CS, Chang PC, Fan KC (2006) Motion-embedded residual error for packet loss recovery of video transmission and encryption. In: Proceedings of the IS&T/SPIE: visual communications and image processing (EI127). Published in SPIE. Proceedings Vol. 6077, 1–14

Thanh TM, Iwakiri M (2014) A proposal of digital rights management based on incomplete cryptography using invariant Huffman code length feature. Multimedia Syst 20(2):127–142

Trappe W, Wu M, Wang ZJ, Liu KJR (2003) Anti-collusion fingerprinting for multimedia. IEEE Trans Signal Process 51:1069–1087
Tzeng J, Hwang WL, Chern IL (2005) An asymmetric subspace watermarking method for copyright protection. IEEE Trans Signal Process 53(2):784–792
Wang MS, Chen WC (2007) Digital image copyright protection scheme based on visual cryptography and singular value decomposition. Opt Eng 46(6):1–8
Wang SH, Lin YP (2004) Wavelet tree quantization for copyright protection watermarking. IEEE Trans Image Process 13(2):154–165
Wen J, Severa M, Zeng W, Luttrell MH, Jin W (2002) A format-compliant configurable encryption framework for access control of video. IEEE Trans Circ Syst Video Technol 12(6):545–557
Wu M, Trappe W, Wang ZJ, Liu KJR (2003) Collusion-resistant fingerprinting for multimedia. IEEE Signal Process Mag 21(2):15–27
Xu X, Dexter S, Eskicioglu AM (2004) A hybrid scheme of encryption and watermarking. In: IS&T/SPIE symposium on electronic imaging 2004, security, steganography, and watermarking of multimedia contents VI conference Vol. 5306. pp 725–736
Zeng W, Liu B (1999) A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images. IEEE Trans Image Process 8(11):1534–1548
Zhao H, Liu KJR (2006) Fingerprint multicast in secure video streaming. IEEE Trans Image Process 15(1):12–28
Zhu BB, Yuan C, Wang Y, Li S (2005) Scalable protection for MPEG-4 fine granularity scalability. IEEE Trans Multimed 7(22):222–233