Multimedia Systems
DOI 10.1007/s00530-015-0472-7

REGULAR PAPER

Fragile watermarking with permutation code for content-leakage in digital rights management system

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

Received: 30 September 2014 / Accepted: 12 June 2015

Abstract In this paper, we present a new scheme of digital rights management (DRM) system employing the fragile watermarking with permutation code for the image distribution via network. General DRM systems are designed to protect the copyright of contents and to trace the source of the illegal distributors based on the user-side watermarking. However, in the typical DRM systems, the original digital contents are temporarily disclosed without the watermarking information inside user's system by the decryption process. Therefore, the user can copy the leaked original content inside the system and illegally redistribute via network without the permission of the content providers. Our work describes the idea of a DRM method which is composed of the incomplete cryptography based on permutation codes and user identification mechanism to control the quality of digital contents. There are two fundamental steps in our proposed cryptography: incomplete encoding and incomplete decoding. These two steps will create the scrambled content that is used as trial content and the watermarked content that is used to prevent unauthorized duplication or business of digital contents, respectively. Experimental results show that the proposed method is suitable for DRM in the network distribution system.

Keywords Digital rights management (DRM) · Incomplete cryptography · Invisible fragile watermarking · Permutation code

1 Introduction

1.1 Background

With the popularity of the Internet and the development of digital multimedia technology, there has been an explosion in the use of digital media through e-commerce business and online services. Since the digital media are easily reproduced and manipulated, everyone can easily copy and illegally redistribute the digital content via network. It is hard to prevent the illegal uses without the copyright rights management technique. Thus, the need for an effective rights management system, where only the legitimate consumers can access to the digital content, is required recently [1–6].

In recent DRM systems [7–9], the fingerprinting information such as user's information is embedded into the content to prove from illegal users or to trace the source of pirated copies. However, the conventional DRM technologies are separately manipulated by the encryption and the watermark method. Therefore, the original content is disclosed temporarily inside the system in the user's decryption (key management process) [10]. In that case, the legal users can save the original contents without the watermark information and unauthorizedly distribute it via network. This is the problem of the conventional DRM systems because the digital content may be redistributed from the

Communicated by L. Zhou.

✉ Ta Minh Thanh
taminhjp@gmail.com; thanh4@is.titech.ac.jp

Munetoshi Iwakiri
iwak@nda.ac.jp

[1] Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-2, Ookayama, Meguro-ku, Tokyo 152-8552, Japan

[2] Department of Network Security, Le Quy Don Technical University, 236 Hoang Quoc Viet, Hanoi, Vietnam

[3] Department of Computer Science, National Defense Academy, 1-10-20, Hashirimizu, Yokosuka-shi, Kanagawa 239-8686, Japan

Published online: 30 June 2015

legal users to unauthorized users. In this follows, a user may legally purchase a digital content, however, he/she can illegally distribute it to other users without the owner's permission. This work focuses on to address the above-mentioned problems.

1.2 Related works

There are some existing researches that focused on the dual targets of the digital content access and the traitor tracing [10–15]. They employed the digital watermarking or the fingerprinting for copyright protection of the digital content. Generally, the watermarking is applied towards copyright protection, but the fingerprinting is used for traitor tracing. In the literature, the watermarking is employed at the server-side system [16], in which, an unique watermark information is embedded in all digital contents. Therefore, when the producer detected the illegal redistribution via network, although the watermark is extracted from the suspected content, he/she cannot specify the traitor. On the other hand, the fingerprinting is normally applied at the user-side [13–15] because one digital content will have many different users. The user information is used to represent different users' identity. Therefore, when the producer detected the illegal redistribution via network, he/she can easily specify the traitor.

According to the above analytics, the fingerprinting is the promising technique for digital content access and traitor tracing. Server-side encryption and user-side fingerprint embedding (conventional DRM system) was first proposed in [13] and then was extended by [14, 15]. The concept of this model is described in Fig. 1. In this scenario, only one global key based encryption is necessary at the server side. Then, the encoded content is sent to different users via network by multicasting. At the user-side, the encoded content can be decrypted by using the global key. Here, a watermarked software (DRM controller software) is necessary for combining the content decryption and the 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).

In order to solve the original content leakage problem, Kundur and Karthik [11, 12] proposed a joint fingerprinting and decryption (JFD) method, which is shown in Fig. 2. The idea of JFD is that the encoded content is partially decrypted in which some un-decrypted parts are remained to imitate the fingerprint embedding. Therefore, JFD can eliminate the original content leakage problem and help to prevent a traitor from obtaining the decrypted content without watermark information. However, JFD must satisfy the following two conflicting requirements. The first one is that the un-decrypted parts should not degrade the quality of the fingerprinted content. The second one is that the un-decrypted parts should ensure the meaning of the encryption that completely hides their original parts.

With the similar motivation, Chameleon method was proposed by Anderson and Manifavas [17] based on secret table look-up operations in order to prevent the content leakage problem. In their paper, the watermarked contents are decrypted using different secure tables according to users. Therefore, the Chameleon method can distinguish different users by checking the fingerprint for each secure table. However, the Chameleon method may consume greater bandwidth and time-consuming in checking process because each user needs a different secure table [18].

Based on another idea, Lin et al. [10] proposed the fingerprinting method and user-side using vector quantization domain (FVQ). FVQ employs the permutation and the codeword substitution tables using static key-trees or dynamic key-trees. It can save significant bandwidth and conveniently update key-trees. However, in FVQ scheme,

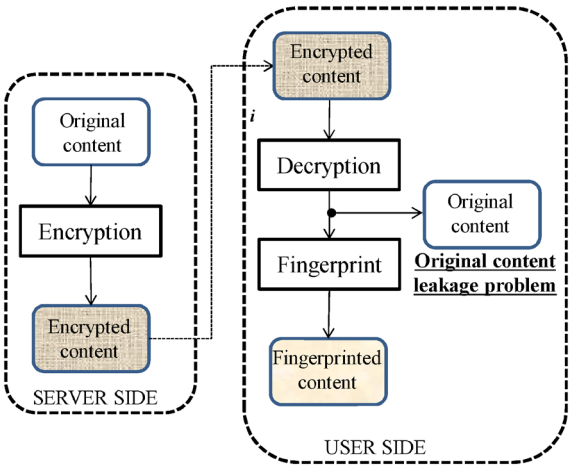

Fig. 1 Server-side encryption and user-side fingerprint method

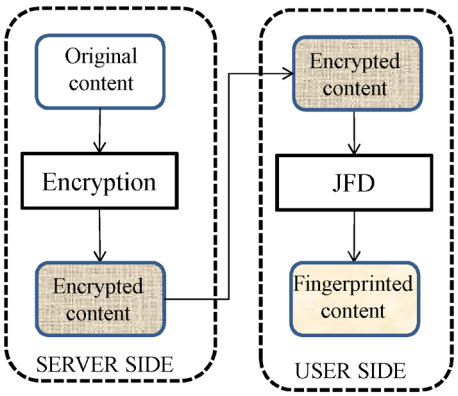

Fig. 2 Joint fingerprinting and decryption (JFD) method

there is no trial content for users try it before deciding to purchase it.

1.3 Challenging issues

Based on the related works, we summarize the challenging issues as follows:

- *Issue 1* Prevention of the original content leakage problem Because the general DRM systems are implemented the decryption and watermarking process separately, the original content (without the watermarking information) can be temporarily saved inside the user's system [10, 13–15]. If the user copies the original content and illegally redistributes it via network, the producer cannot detect the illegal re-distributor in this case. The first challenging issue is how to improve the DRM technique in order to prevent the original content leakage problem. Although JFD can momentarily resolve the content leakage problem, however, JFD still exists some drawbacks such as two conflicting requirements that mentioned before.
- *Issue 2* Detection of the illegal re-distributor As previously mentioned, some DRM techniques [10–12, 17], which combine the encryption and the fingerprinting technology, have also been proposed. Almost those techniques employed the different secure tables or the codeword substitution tables that are decided by the producer to detect the illegal re-distributor. However, the detection method using secure tables/codeword tables always consumes greater bandwidth and it seems to be complex. Hence, the second challenging issue is how to implement simply detection scheme in the DRM system.
- *Issue 3* Judgement of the illegal user Previous works normally uses the watermark or the fingerprint information that is predefined by the producer to judge the illegal users [21, 22]. It means that when the producer can extract the watermark information from the suspected content, the user who possesses the content will be regarded as the legal user. However, the illegal users (the unauthorized users) can posses the content by purchasing it from the legal user without the permission of the producer. In this case, although the illegal users are not authorized by the producer but they can be judged as the legal user because the watermark information can be extracted from their content. Therefore, the third challenging issue is how to propose new scheme to judge the illegal user when the suspected content is detected.
- *Issue 4* Usage of trial content We consider that the trial contents in the DRM system play very important role to advertise the content widely via network. By using the trial content beforehand, the user can make the decision whether to purchase the content or not. Unfortunately, most aforementioned existing DRM did not exploited the trial content inside the flow of system. Only JFD [11] proposes the encrypted content that can be used as trial content in the flow of the DRM system. The last challenge issue is how to provide the trial content.

1.4 Our contributions

This paper describes new design and implementation of the DRM technique based on the incomplete cryptography system by using the fragile watermarking with permutation code. The incomplete cryptography is proposed for improving the problem of conventional DRM system. Our method will deteriorate the quality of the original contents to make the trial contents for distribution to users via network. The quality of the trial contents will be controlled with the watermarked keys at the incomplete decoding process, and the user informations will be embedded into the incomplete decoded contents simultaneously. Therefore, our technique can resolve the Issue 1 by combining two processes (decryption and fingerprinting) at the user-side to become the incomplete decoding. Based on this, the individual user's information is embedded into the decoded content during the decoding process.

In our system, we ask the users to register his/her information at their purchasing process. We use their information as the watermark information to create the decoding key for individual user. Therefore, when a legal user uses the decoding key including his/her own information to decode the trial content, the user's information is embedded into the decoded content. If a legal user redistributes his/her own content via network, the producer can extract the watermark information from the suspected content and can detect the re-distributor based on his/her own information (resolving the Issue 2).

To address the Issue 3, we also propose new scheme for judgement of illegal user using fragile watermarking information that is extracted from the suspected content. The legal user can be judged only in case the user's information is extracted from the content. Therefore, if the fragile watermarking information is not matched with user's information, the user is judged as an illegal user. Also, if the legal user tries to convert the content to another version by using image processing method or geometrical processing method to redistribute via network, of course, the fragile watermarking information can not be extracted, the users use such kind of content will be judged as the illegal user.

Also, we propose the technique to provide the trial content (Issue 4) to advertise the content widely via network to users. Our technique can create the scrambled content and upload it to internet for users. It helps user easily to make the decision on purchasing the digital content.

Finally, our proposed method in this paper is suitable for the application of image distribution via the Internet. We can employ the components of the image in order to adjust the quality and to embed the fragile watermark information. In case of the JPEG image, the DCT coefficients are the important component for controlling the quality of itself. Therefore, we use the JPEG algorithm to apply our proposed method and prove its efficiency in the real applications. We apply the proposed method on the JPEG image because JPEG is the most common used image format. Especially, JPEG format is preferred to be used in the digital cameras.

The rest of the paper is organized as follows: In the Sect. 2, we briefly explain the review of the incomplete cryptography system. Inspired by it, we provide a new model of DRM system based on the incomplete cryptography. The implementation of incomplete cryptography using permutation codes is explained in Sect. 3. Experimental results on the JPEG algorithm are presented in Sect. 4 to demonstrate the performance of the proposed method in the network distribution system. Finally, we conclude this paper in Sect. 5.

2 The idea of incomplete cryptography

2.1 Outline

The proposed incomplete cryptography [19, 20] consists two steps: the incomplete encoding and the incomplete decoding (see Fig. 3).

In the incomplete encoding process, the original content P is encoded based on the encoder function E with the encoder key k to make the scrambled content C.

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

Here, C can be simply recognized as a part of P (even if C is not decoded). This feature is called *incomplete confidentiality*.

On the other hand, the incomplete decoding process is different from the complete decoding process. C is decoded by using a decryption function $D' \neq D$ and a decoded key $k'_i \neq k, i = 1, 2, \ldots, N_u$ to create P'_i, where N_u is the number of the legal users.

$$P'_i = D'(k'_i, C). \tag{2}$$

Since P'_i is decoded by another decryption function D' with key k'_i, it will be different from the original content P. Therefore, the relationship of P and P'_i is $P'_i \neq P$ in the incomplete cryptography system. This feature is called *incomplete decode*.

The main contribution of the incomplete cryptography is that the quality of P'_i can be controlled with a particular key k'_i. When C is decoded with k'_i, P'_i is not only decoded with slight distortion, but also is watermarked with an individual user information that is used as fingerprinting information. It is the elemental mechanism of fingerprinting based on the incomplete cryptography system.

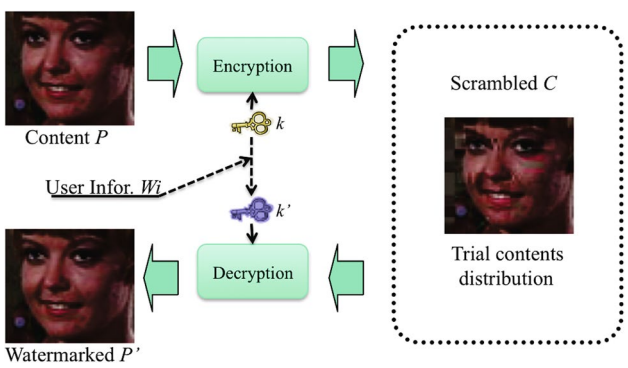

Fig. 3 The incomplete cryptography

2.2 The design of DRM system based on incomplete cryptography

The idea of the DRM system based on the incomplete cryptography is explained in this section. A DRM system requires to deliver the original contents to legal users safely and smoothly. When a DRM system is constructed by using the incomplete cryptography, it is not only the safety distribution method to users, but also the solution for the conventional DRM problem.

Before distribution, the producer T has a digital content P and needs to be sent to users as much as possible. Thus, T creates a scrambled content C with the encryption key k based on the incomplete cryptography. Then, C is to disclose a part of P. It means that C is maintained over the minimum quality of P. T distributes C to users widely via network as the trial content.

After trial of C, the user R_i decides to purchase the digital content. Then, R_i has to register his/her individual information. This information will be used as the watermarked information (W_i) and it will be embedded into the content. When T receives the purchaser's agreement of R_i, T sends a watermarking key k'_i to the user R_i. k'_i is the incomplete decoding key and it is prepared individually to each user.

R_i decodes C using k'_i and obtains the high quality content P'_i. In this incomplete decoding process, ID information (W_i) of user will be embedded in P'_i as the watermarking information.

Therefore, when a producer wishes to check whether a user is a legal user, he/she can extract the watermarking information from P'_i and compare with his user database. If the watermarking information matches his database, the user is legal. Conversely, the user is illegal. Furthermore, it can specify to trace the source of pirated copies. The purpose of this proposed method is to inform the user about

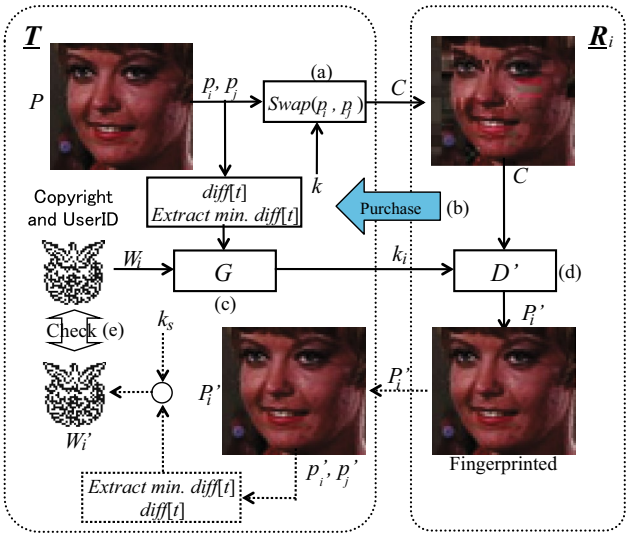

Fig. 4 Incomplete cryptography using permutation codes

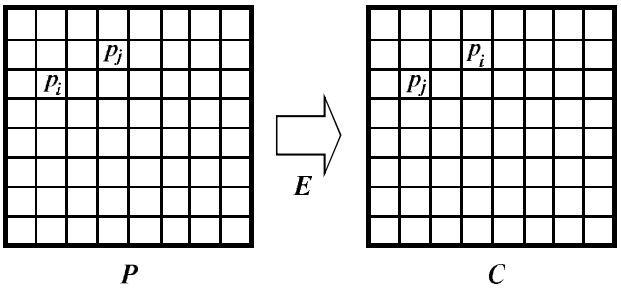

Fig. 5 Swapping position of permutation codes

the existence of watermarking which can exactly identify the users. That can limit the illegal redistribution in advance.

3 The proposal of incomplete cryptography using permutation codes

In this section, firstly, we explain the mechanism to create the scrambled content for the trial content by using the permutation codes method. Secondly, an algorithm for making the watermarking key is presented. Then, the incomplete decoding process with the watermarking key in which the user information is embedded into the decoded content, is explained. Figure 4 shows the process of the proposed permutation codes method.

3.1 Incomplete encoding

The scrambled content is the result of the incomplete encoding. This procedure is explained as follows:

The producer T randomly selects the specified coefficients p_i and p_j from the original content $P = \{p_0, p_1, \ldots, p_i, \ldots, p_j, \ldots\}$ by using the encoded key k. The coefficients p_i and p_j are swapped by $Swap(p_i, p_j)$ function to generate the scrambled content C (Fig. 4a). After swapping, the positions of p_i and p_j are exchanged as shown in Fig. 5.

$$C \leftarrow Swap(p_i, p_j), \tag{3}$$

where i, j denotes the coordinates of the selected coefficient, $i, j \in [1, 2, \ldots, N \times N]$, $i \neq j$, and $N \times N$ denotes the block size of P.

In our incomplete encoding process, we only swap some specified portions of the coefficients in the original content. Therefore, when we apply $Swap(p_i, p_j)$ function for scrambling the content $P = \{p_0, p_1, \ldots, \underline{p_i}, \ldots, \underline{p_j}, \ldots\}$, we can obtain the scrambled content $C = \{p_0, p_1, \ldots, \underline{p_j}, \ldots, \underline{p_i}, \ldots\}$. In order to swap the multiple coefficients in P, we choose L pairs of the coefficients $\{p_i^t, p_j^t\}, t = 1, 2, \ldots, L$ to encode P.

Since only some significant coefficients are swapped, the scrambled content C is degraded to lower quality that is suitable for the trial content. After encoding P, C is delivered widely to users R_i via network.

3.2 Management of the individual user information

Suppose that an user R_i decides to purchase a content P. He/she needs to register his/her individual information as the license copyright (LC) of the legal user. R_i can register his/her own logomark or user information such as birthday, telephone number. T employs user's logomark or assigns the diverse watermark for each LC in order to generate the individual watermarking key. Using the LC, T can find out the legal user who bought the digital content. In this work, we use a binary picture W_i (Nda32) (Fig. 9d) with size $M \times M$.

In order to generate the watermarking key, we extract the bit sequences from W_i. $W_i(x, y)$ is converted into a linear array $W_l(z) = W_i(x, y), \quad z = x + yM, \quad 1 \leq x, \quad y \leq M$. One bit $b_l \in W_l(z)$ is used to generate the watermarking key as the process of Sect. 3.3.

To reconstruct the watermark W_i', the bit sequences $W_l'(z)$ is collected from all extracted bits b_l' in Sect. 3.5. Two dimensional watermark $W_i'(x, y)$ is formed from $W_l'(z)$ as $W_i'(x, y) = W_l'(z), z = x + yM, 1 \leq x, y \leq M$.

3.3 Generation of watermarking key

In our proposed system, to decode the scrambled content C, a decoding key k_i' is required from a user R_i after purchasing. k_i' is created by the generation function G based on the individual user information W_i that is registered in purchasing process (Fig. 4b). k_i' is also the watermarking

key because of using the k'_i, the user information W_i will be embedded into the decoded content at the locations that are specified by k'_i. Therefore, the original content is not disclosed temporarily inside the system in the decoding process. Furthermore, our algorithm is considered that the quality of decoded content P'_i is controlled by the watermarking key k'_i. It means that k'_i is used to minimize the degradation of decoded content (watermarked content).

This section describes the watermarking key generation algorithm based on selective differential method (Fig. 4c). The steps of selective differential method are described as follows:

Step 1. Obtain the absolute difference $diff[t]$ of L pairs $\{p_i^t, p_j^t\}$ are given by,

$$diff[t] = |p_i^t - p_j^t|. \tag{4}$$

Step 2. Find the minimum $min(diff[t])$. Record the coordinates of the pair $\{\bar{p}_i^t, \bar{p}_j^t\}$ that has the $min(diff[t])$ and suppose those coordinates are (x_1, y_1) and (x_2, y_2), respectively. The coordinates (x_1, y_1) and (x_2, y_2) are locations to embed one bit of the user information. They are registered as the secret key k_s. In our algorithm, (x_1, y_1) and (x_2, y_2) are selected so that the two coefficients have the smallest value of $diff[t]$ to preserve the quality of the images.

Step 3. Extract one bit b_l from user information $W_l(z)$ (see Sect. 3.2) and generate the individual fingerprinting key k'_i as follows:

$$k'_i = G(\bar{p}_i^t, \bar{p}_j^t, b_l). \tag{5}$$

When $b_l = 0$:

$$k'_i = \begin{cases} \text{no change} & \text{if } \bar{p}_i^t > \bar{p}_j^t, \\ Swap(\bar{p}_i^t, \bar{p}_j^t) & \text{otherwise.} \end{cases}$$

When $b_l = 1$:

$$k'_i = \begin{cases} \text{no change} & \text{if } \bar{p}_i^t < \bar{p}_j^t, \\ Swap(\bar{p}_i^t, \bar{p}_j^t) & \text{otherwise.} \end{cases}$$

Step 4. Excepting the coordinates (x_1, y_1) and (x_2, y_2) of the pair that is specified in Step 2, other coefficients, $(L-1)$ pairs, which has the coordinates are not recoded in k_s will be completely decoded by the function $Swap(p_i^t, p_j^t)$.

According to above algorithm, the decoded key k'_i is generated based on the individual user information. k'_i is also expected that it can control the quality of the decoded content P'_i in the decoding process with minimum distortion.

In order to avoid the inspection of the adversary, our method can randomly choose the DCT tables inside the JPEG image to embed the user information. It ensures that the existence of the watermark may not be noticed by the adversary.

3.4 Incomplete decoding

The watermarked content P'_i is created by the user R_i at the receive side. After purchasing, R_i receives a watermarking key k'_i from T. R_i uses k'_i to decode the scrambled content C and to obtain the watermarked content P'_i (Fig. 4d). In the decoding process, the individual information of R_i is embedded synchronously as fragile watermarking information.

$$P'_i = D'(k'_i, C). \tag{6}$$

Note that, because of P'_i is decoded by another secret key $k'_i (\neq k)$ with the decoder function D', P'_i is different from the original content P. Therefore, the relationship of P and P'_i is $P'_i \neq P$ in the incomplete cryptography system.

Thus, T can control the quality of P'_i (watermarked contents) with the particular key k'_i (watermarking key). Then, when the user decodes C using k'_i to achieve P'_i, P'_i is not only decoded with slight distortion, but also is watermarked with the individual user information.

3.5 Extraction of user information

Assuming that there is a digital content P'_i, which is redistributed widely via network. A producer discovers and wishes to check who illegally redistributes the digital content. He/she can extract the user information from the redistributed content, then compare with his/her database to detect the unauthorizedly redistributed user.

In the watermarking information extraction, the embedded information bit sequences can be detected by comparing significant coefficients $\{\bar{p}_i^t, \bar{p}_j^t\}$ that is specified by the secret key k_s (Fig. 4e).

$$b'_l = \begin{cases} 0 & \text{if } \bar{p}_i^t > \bar{p}_j^t, \\ 1 & \text{otherwise.} \end{cases} \tag{7}$$

After extracting the user information from the redistributed content, the producer can easily specify to trace the source of pirated copies. If the user information matches with his database, the user is legal. Conversely, the user is illegal.

Note that, in our proposed method, we define that if the user information is not detected or is not matched with the database of the producer, the user who possessed the suspected content will be judged as a illegal user. Therefore, if a legitimate user tries to remove/replace his information by another watermark before redistribution, such contents are considered as the illegal content. Also, if the legal user converts the content to another version by using intentional

attacks such as image processing methods or geometrical processing methods to redistribute via network, of course, the user information cannot be extracted. If the user uses such converted content, he/she will be judged as an illegal user. Additionally, to avoid the unintentional conversions, we implement our proposed method on JPEG images.

4 Experimental results

In this section, we implement a straightforward incomplete cryptography scheme based on the permutation codes method. Based on the proposed method, we implement the proposed DRM system exploited the standard JPEG algorithm [23].

4.1 Summary of JPEG algorithm

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

To make the scrambled and incomplete decoding contents of JPEG, and we have selected the quantized DCT coefficients to implement the incomplete cryptography. There are two reasons for this choice. First, it is easy to control the image quality by adjusting the quantized DCT coefficients. The second reason is the flexibility of making a variation of content by selecting the luminance component (Y component) and two chrominance components (UV component) in the quantized DCT coefficients.

4.2 Environment and evaluation

All experiments are performed by the incomplete encoding and the incomplete decoding on the JPEG images using the Vine Linux 3.2 system. In order to generate the encryption k, we use function *rand*() of the 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.

We implement our method on grayscale and color images to confirm the efficiency. We use 50 different 512×512 grayscale images as the test images [25]. Besides, we prepare some different features of the color experimental images regarding CG, scenery, construction and person. Ten test images are the 8-bit RGB images of SIDBA (Standard Image Data BAse) international standard image (Lighthouse, Pepper, Title, Lena, Girl, Airplane, Parrots, Couple, Milkdrop, Mandrill) of size 256×256 pixels. We use the additional database images ISO/JIS-SCID (Party, Picnic, Portrait)[3] with 2048×1536 pixels, 8-bit RGB. Here, all images are compressed with quality 75 (the lowest $0 \leftrightarrow 100$ the highest) to make the experimental JPEG images for evaluation of the proposal method.

We prepare a bitstream 32×32 pixels of binary picture (Nda32) as watermarking information (see Fig. 9d).

4.3 Evaluation of image quality

We use PSNR (Peak Signal to Noise Ratio) [24] to evaluate the JPEG image quality. The PSNR of $H \times W$ pixels image of $g(i,j)$ and $g'(i,j)$ is calculated with,

$$PSNR = 20 \log \frac{255}{MSE} \quad [\text{dB}],$$
$$MSE = \sqrt{\frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \{g(i,j) - g'(i,j)\}^2}, \quad (8)$$
(MSE : mean square error).

The structural similarity index measure (SSIM) [27] is also used as criteria to estimate the invisibility of the processed images. When the value of SSIM is close to 1, that implies the quality of image is indistinguishable from the original images.

In these experiments, the PSNRs are calculated with RGB pixel data of the original image and the JPEG image. A typical value for PSNR in a JPEG image (quality 75) is about 30dB [24].

We also employ the MOS (Mean Opinion Score) experiments to evaluate the experimental images. First,

[1] http://gcc.gnu.org/.
[2] http://www.imagemagick.org/script/.
[3] http://www.colour.org/tc8-04/Sony_sRGB_Standard_Image_1999/.

we evaluate the relationship of the subjective image quality and PSNR. Here, we prepare ten images of 15-32dB of PSNRs. Those are controlled with the DCT coefficients of Y component and those of UV component, respectively. Afterward, the experimental JPEG images are assessed subjectively with ten testers and the MOS values are calculated. Ten testers are the students in our lab at $18 \sim 25$ ages.

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 is also reported as perceived quality of test JPEG images as, 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".

In MOS experiment, Lena, Lighthouse, Pepper and, Title are used as test images. At the beginning of the MOS test, the tested JPEG images are randomly shown to testers. The MOS grades of them is not evaluated until they provide stabilization of the perception of tester.

Figure 6 shows the relation between the MOS and the PSNR. According to Fig. 6, we realize that the testers feel the deterioration when the value of PSNR of image is lower than approximately 22dB (MOS: 0–2.5). In addition, when the PSNR is between 22dB and 28dB (MOS: 2.5–3.5), testers feel the deterioration but slightly annoying. The image quality in this case is considered to be acceptable for the scrambled content. When the PSNR is higher than 29dB (MOS: 3.5–5), testers almost cannot feel the deterioration of image.

Based on the MOS experimental results, we conclude that the PSNR of the scrambled image is appropriately between 22dB and 28dB. The PSNR of the decoded image should be higher than 28dB.

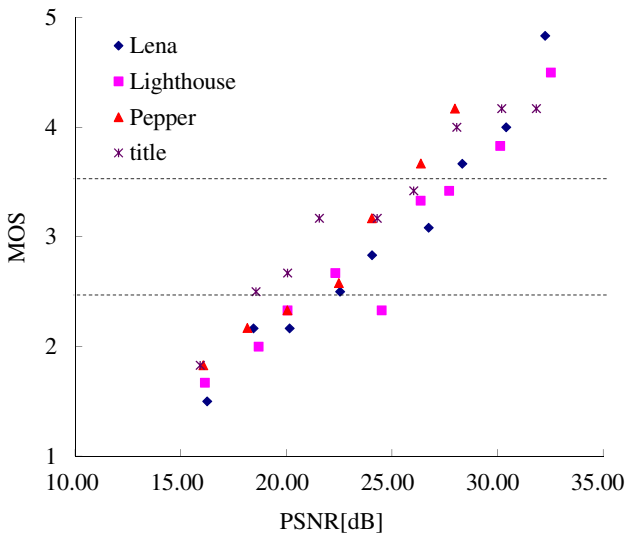

Fig. 6 The correspondence of MOS and PSNR

4.4 Evaluation of watermark

To evaluate the similarity of the reconstructed binary watermark $W_i'(x, y)$ with the original one $W_i(x, y)$, we calculate the normalization correlation (NC) value of the those as follows,

$$NC = \frac{\sum_{x=1}^{M} \sum_{y=1}^{M} [W_i(x, y) \times W_i'(x, y)]}{\sum_{x=1}^{M} \sum_{y=1}^{M} [W_i(x, y)]^2}. \tag{9}$$

If NC value is close to 1, it means that the reconstructed binary watermark is similar to the original one.

Since the size of the binary watermark is smaller than that of the original JPEG image, we repetitively embed the watermark into the original image. Therefore, multiple binary watermarks can be extracted from the watermarked image. In case of multiple binary watermarks are extracted, we use the voting method in [26] to derive the most accurate binary watermark before calculating the NC value.

4.5 Watermarking capacity

In our proposed method, the watermarking capacity is proportional to the number of partitioned blocks. We can calculate the watermarking capacity for the Y component and the UV component based on the size $H \times W$ of original image. In case of the JPEG image, the block size is 8×8. Moreover, since we use JPEG image with YUV420 format, therefore, the number of blocks of UV component is equal to half of that of Y component. In our paper, we only embed one bit watermark into one block.

If we employ the Y component for watermark embedding, the watermarking capacity *Emb* is calculated as follows:

$$Emb = \frac{H}{8} \times \frac{W}{8} \text{ [bits]}. \tag{10}$$

If we employ the UV component for watermark embedding, the watermarking capacity *Emb* is calculated as follows:

$$Emb = \frac{1}{2} \times \frac{H}{8} \times \frac{W}{8} \text{ [bits]}. \tag{11}$$

The total of watermarking capacity *Emb* when the Y component and the UV component are employed, can be calculated as follows:

$$Emb = \frac{H}{8} \times \frac{W}{8} + \frac{1}{2} \times \frac{H}{8} \times \frac{W}{8} \text{ [bits]}. \tag{12}$$

4.6 Simulation results

This Section presents some empirical results concerning the incomplete encoding and the incomplete decoding on JPEG images using the permutation codes method.

4.6.1 Example of implementation

First, the quantized DCT tables (size 8×8, $N = 8$) are extracted from JPEG image. Then, the proposed permutation codes method is applied to each DCT table of the Y component and the UV component.

The details of these experimental methods are shown in Fig. 7. In order to make a scrambled content, the quantized DCT table P is extracted from the JPEG image. An encryption key k is generated to specify the random L pairs $\{p_i^t, p_j^t\}$ in P.

For instance, Fig. 7a is a part of the DCT table that is selected to make the scrambled content. Assuming that, three pairs ($L = 3$): $\{(6,2),(4,2),(5,2)\}$ are specified according to the encryption key k. The locations of each coefficient in these pairs are swapped to make the scrambled content C. The results of swapped method are $\{(2,6),(2,4),(2,5)\}$ as shown in Fig. 7b. The quality of C can be controlled by degrading the quality of P based on handling the number of the swapped pairs. Note that, the quality of C is needed to be suitable for the trial content. The producer will control the quality of the trial content around 20dB.

On the other hand, to prove the efficiency of watermarking key generation method using selective differential algorithm, we try to generate the fingerprinting key k_i' to decode the scrambled content. As the decoding process, two pairs are completely swapped again to restore the original locations. The remaining of pair is devised for embedding the watermark bit. In example of Fig. 7b, we obtain the absolute difference of three pairs and get the results $diff[t] = \{4, \underline{2}, 3\}$. From these results, it is clear that "2" is the minimum of $diff[t]$. Thus, the location of minimum pair is specified location of the fingerprint bit. Therefore, $(x_1, y_1) = (1,1)$ and $(x_2, y_2) = (0,2)$. If fingerprint bit is $b_l = 0$, the location of the selected pair "(2, 4)" is not changed and is remained as the location in the scrambled content (see Fig. 7c1). Otherwise, if the fingerprint bit is $b_l = 1$, then the location of the selected pair "(2, 4)" is swapped as Fig. 7c2. With respect to the selection of the embedded location in selective differential method, our incomplete decoding process is expected to minimize the distortion of the decoded content after fingerprinting and decoding process.

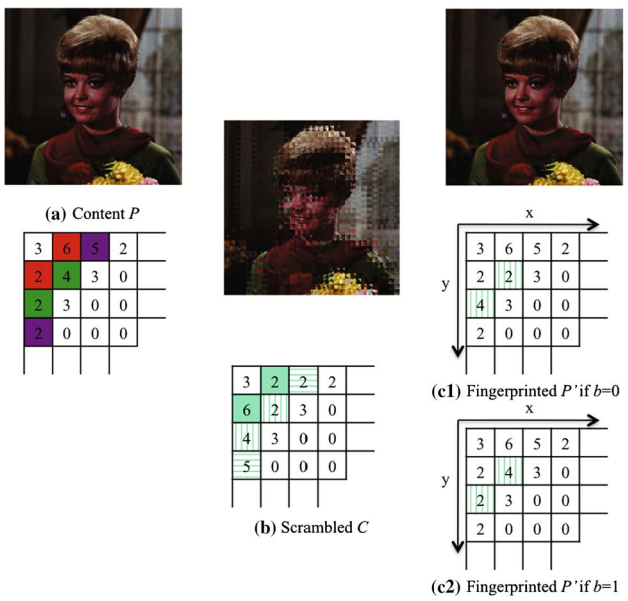

Fig. 7 An example of implementation permutation codes in DCT table

To extract the user information from the fingerprinted JPEG image, it can be extracted from location (x_1, y_1) and (x_2, y_2) of P_i' by applying the same selective differential algorithm and compare with userID in database to confirm the illegal user.

4.6.2 Results of grayscale images

Since JPEG images of grayscale images have only Y component, we just apply our proposed method for only Y component of those images. In the experiments, 1024-bits of Nda32 logo is embedded into the 50 JPEG grayscale images. Figure 8 shows the PSNRs of the JPEG grayscale images. Obviously, the efficiency of Y component in grayscale JPEG images is extremely strong because there are some decoded images had low quality (PSN < 30dB) after embedding the user's information. In order to improve the quality of the decoded JPEG images, we can reduce the amount of user's information bit to control the quality of the decoded images.

4.6.3 Results of color images

Unlike the grayscale JPEG images, color JPEG images consists of three components: luminance component (Y component) and chrominance components (UV component). We also conduct our proposed method on those to confirm the efficiency of system. We embed the logo into all DCT tables of the Y component or that of the UV component. Therefore, total 1024-bits and 512-bits are embedded into the Y component and the UV component, respectively.

The experimental results are shown in Table 1. We can see that the watermarked JPEG images are indistinguishable from the original JPEG images. We calculate the PSNR value of the output JPEG images in every processes and extract the watermark information (embedded binary watermark) perfectly from the incomplete decoded JPEG images. Obviously, the scrambled JPEG images are degraded about 20dB, and they seem to be appropriate as the trial content. The PSNRs of the decoded JPEG are higher than 28dB. That implies that our watermarking

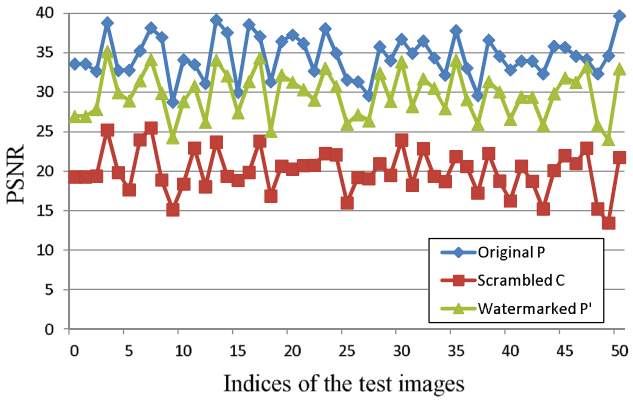

Fig. 8 PSNRs of the JPEG grayscale images

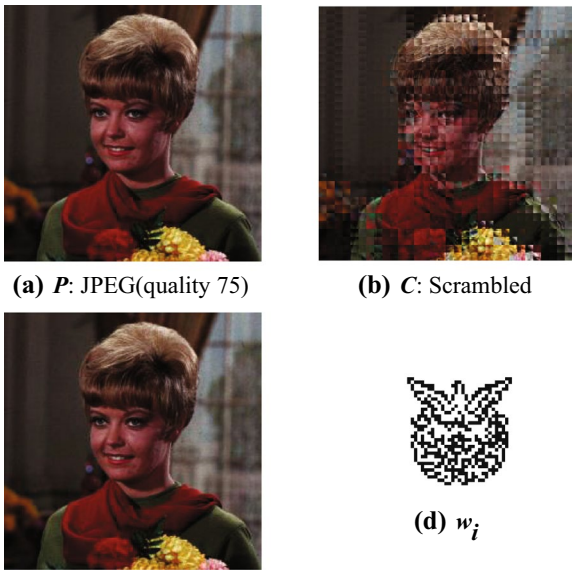

Fig. 9 Examples of Girl images

scheme in the incomplete decoding process can achieve visual transparence.

The SSIM values also are calculated for each JPEG images. The SSIM values of the original JPEG images and that of the decoded images are nearly close to 1. That means the quality of the decoded JPEG images is high. Otherwise, the SSIM values of the scrambled JPEG images are far from 1. That means the quality of the scrambled JPEG images is low.

Most notably, our NC values equal to 1. That means, the reconstructed watermarks from the decoded JPEG images are similar to the original watermark. Therefore, it can be recognized clearly by human eyes.

Figure 9 is an experimental sample of the Girl images. According to the results in Fig. 9, it is possible to produce the scrambled content (see Fig. 9b) and the incomplete decoded content (see Fig. 9c) based on the incomplete cryptography using permutation codes method. Furthermore, the watermark can be extracted accurately (see Fig. 9d).

We also compare the results of the UV component and the Y component as shown in Table 1. We confirm that when we manipulate the Y component, the distortion of image is more conspicuous than that when we apply to the UV component. Therefore, we can make the scrambled content efficiently with the Y component. However, because the image deterioration is not conspicuous when implementing the UV component, there is an advantage to embed the abundant watermark information into decoded content under the maintaining its quality. Several results of SIDBA images using the Y component permutation codes method are shown in Fig. 10. These images show that the proposed method is very appropriate for the digital image distribution system.

Table 1 also 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 into the image.

According to the above results, we can establish the incomplete cryptography system based on the permutation codes method. The scrambled content is created to disclose the original content and is distributed widely to users. In the incomplete decode process, we remain the locations of the selected DCT pair to embed the user information by a watermarked key. Thus, the original content is not decoded temporarily inside the system. Therefore, we conclude that above technical problem of the conventional DRM system is solved by using the incomplete cryptography system.

4.7 Comparisons

4.7.1 Comparison of our method with the related works

According to analytics of the related works, JFD [12] seems to be promising technique to achieve decryption and fingerprint embedding at the same time. However, since un-decrypted parts in JFD are employed as the fingerprinting information for user, then the quality of the decrypted content is limited. Our method uses the user ID that is embedded into the specified position, therefore, our method can flexibly control the quality of the decrypted content. In addition, in our method, the incompletely decrypted blocks (watermarked blocks) instead of un-decrypted blocks, therefore, our method can take better trade-off between multimedia security and fingerprinting imperceptibility than JFD.

Fig. 10 Example images of SIDBA dataset using Y component permutation codes method. The *first row* is original images, the *second row* is scrambled images and the *third row* is fingerprinted images

Compared to FVQ [10], our method can provide the trial contents to users via network. The users can try it before deciding whether to purchase content or not. Besides, the encryption key and decryption key of our method seem more effective than FVQ with user-based information, i.e., UserID and our method also can detect the traitor of the suspected digital contents. The detail of comparison of our method with JFD and FVQ is shown in Table 2.

4.7.2 Capacity

We also compare our embedding capacity to method [20]. In [20], we proposed three methods using the feature of the Huffman code of the JPEG codec to embed the user information. Three methods are IHAF: Invariant Huffman code length AC coefficient Fingerprinting; IHDF: Invariant Huffman code length DC coefficient Fingerprinting; IHOF: Invariant Huffman code length Offset AC coefficient Fingerprinting. However, the capacity of the embedded bits, is limited by the Huffman table. By using the proposed method, the capacity is effectively improved. The results of comparison are shown in Table 3. We realize that the capacity of our proposed method is better than [20] and it is suitable for the real applications.

5 Conclusions

In this work, we have presented a scheme to implement the incomplete cryptography using the permutation codes. We also have described a design for fingerprinting with Digital Rights Management system. This approach integrated the encoding process and the fingerprinting progress of DRM technology. By doing so, we could eliminate the problem

Table 1 PSNR[dB]/SSIM, embedded bits and NC values

Method	Image	P	C	P_i'	*Emb* [bits]	NC
Y comp.	Airplane	30.20/0.962	17.95/0.822	28.24/0.895	1024	1
	Girl	32.70/0.954	21.90/0.782	31.13/0.920	1024	1
	Parrots	34.26/0.963	21.24/0.882	31.82/0.930	1024	1
	Couple	34.04/0.958	22.37/0.669	31.45/0.905	1024	1
	Pepper	28.81/0.960	18.05/0.883	27.98/0.894	1024	1
	Lighthouse	32.67/0.942	19.83/0.890	31.94/0.938	1024	1
	Lena	32.37/0.959	18.18/0.896	28.17/0.913	1024	1
	Milkdrop	31.99/0.923	20.55/0.833	29.73/0.871	1024	1
	Party	35.16/0.993	20.68/0.823	31.56/0.979	49152	1
	Picnic	34.39/0.993	22.07/0.848	31.55/0.977	49152	1
	Portrait	35.99/0.988	23.50/0.749	33.44/0.977	49152	1
UV comp.	Airplane	30.20/0.962	25.73/0.822	29.48/0.961	512	1
	Girl	32.70/0.954	28.23/0.781	32.42/0.953	512	1
	Parrots	34.26/0.963	23.64/0.883	33.46/0.963	512	1
	Couple	34.04/0.958	29.48/0.668	33.83/0.957	512	1
	Pepper	28.81/0.960	21.62/0.884	28.01/0.959	512	1
	Lighthouse	32.67/0.941	20.83/0.890	31.94/0.941	512	1
	Lena	32.37/0.959	27.82/0.896	32.02/0.958	512	1
	Milkdrop	31.99/0.920	23.81/0.903	30.99/0.919	512	1
	Party	35.16/0.993	29.54/0.823	34.68/0.992	24576	1
	Picnic	34.39/0.993	29.93/0.848	34.18/0.992	24576	1
	Portrait	35.99/0.988	30.43/0.749	35.81/0.988	24576	1
YUV comp.	Airplane	30.20/0.962	17.72/0.821	26.61/0.896	1536	1
	Girl	32.70/0.954	21.24/0.781	30.16/0.920	1536	1
	Parrots	34.26/0.963	19.34/0.882	30.53/0.929	1536	1
	Couple	34.04/0.958	21.97/0.668	30.54/0.899	1536	1
	Pepper	28.81/0.960	16.95/0.883	26.20/0.893	1536	1
	Lighthouse	32.67/0.941	18.18/0.890	27.28/0.878	1536	1
	Lena	32.37/0.959	19.36/0.896	29.35/0.913	1536	1
	Milkdrop	31.99/0.923	19.06/0.833	28.65/0.870	1536	1
	Party	35.16/0.993	20.27/0.823	30.50/0.979	73728	1
	Picnic	34.39/0.993	21.53/0.848	30.61/0.971	73728	1
	Portrait	35.99/0.988	23.11/0.749	32.00/0.977	73728	1

Table 2 Comparison between our proposed method with JFD[12] and FVQ[10]

	Ours	*JFD* [12]	*FVQ* [10]
Domain	Partial encryption/decryption	Partial encryption/decryption	Partial encryption/decryption
Block	Incompletely/completely decryption	Un-decrypted/decrypted	Vector quantization
Coefficient	Watermarked	Un-decrypted/decrypted	Codeword
Fingerprint	User ID	Un-decrypted part	User fingerprinting
Trial content	Yes	Yes	No
Encryption key	One key	One key	Session, trees key
Decryption key	User-based key	User-based key	Session, trees and user-based key

Table 3 Comparison of capacity (embedded bits) with our previous paper [20]

Method	*IHAF*	*IHDF*	*IHOF*	Ours
Airplane	1254	120	1254	1536
Girl	506	252	506	1536
Parrots	604	169	604	1536
Couple	490	179	490	1536
Lena	1012	188	1012	1536
Milkdrop	822	107	822	1536
Pepper	1224	198	1224	1536
Lighthouse	1326	114	1326	1536
Average	904.75	165.88	904.75	1536

of the present DRM technologies and effectively manage the legal users.

One of the lessons had been learned from this paper is that, in order to make the scrambled image and the incomplete decoded JPEG image, it is possible to process the Y component and the UV component flexibly. Also, another lesson is that we can control the quality of incomplete decoded image by using an individual specialized key for each legal user. Subsequently, the fingerprinting information is exactly extracted from the fingerprinted image using our approach. The fingerprinted 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 proposal method is useful for the rights management technology for the content distribution via network.

References

1. Halderman, J.A.: Evaluating new copy-prevention techniques for audio CDs. In: Proceedings of ACM Workshop on Digital Rights Management (DRM), Washington, D.C. (2002)
2. DRM technology: Advanced image seminar 2003. The Institute of Image Electronics Engineers of Japan (2003)
3. Liu, Q., Safavi-Naini, R., Sheppard, N.P.: Digital rights management for content distribution. Australasian Information Security Workshop 2003 (AISW2003), vol. 21, pp. 49–58 (2003)
4. Kim, G.H., Shin, D.K., Shin, D.G.: An Efficient Methodology for Multimedia Digital Rights Management on Mobile Handset. IEEE Trans Consum Electron **50**(4), 1130–1134 (2004)
5. Michiels, S., Verslype, K., Joosen, W., Decker, B.D.: Towards a software architecture for DRM. In: Proceedings of the 5th ACM workshop on Digital rights management, pp. 65–74 (2005)
6. Subramanya, S., Yi, B.: Digital rights management. Potential IEEE **25**, 31–34 (2006)
7. Hartung, F., Ramme, F.: Digital rights management and watermarking of multimedia content for M-commerce applications. In: IEEE Communications Magazine, Selected Papers from ISS2000, pp. 77–84 (2000)
8. Lin, E.T., Eskicioglu, A.M., Lagendijk, R.L., Delp, E.J.: Advances in digital video content protection. Proc IEEE **93**(1), 171–183 (2005)
9. 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)
10. Lin, C.Y., Prangjarote, P., Kang, L.W., Huang, W.L., Chen, T.H.: Joint fingerprinting and decryption with noise-resistant for vector quantization images. Signal Process **92**(9), 2159–2171 (2012)
11. Kundur, D., Karthik, K.: Video fingerprinting and encryption principles for digital rights management. Proc. IEEE **92**(6), 918–932 (2004)
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. Macq, B.M., Quisquater, J.J.: Cryptology for digital TV broadcasting. Proc IEEE **83**(6), 944–957 (1995)
14. Hartung, F., Girod, B.: Digital watermarking of MPEG-2 coded video in the bitstream domain. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 4, pp. 2621–2624 (1997)
15. Bloom, J.: Security and rights management in digital cinema. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 4, pp. 712–715 (2003)
16. Guo, J.M., Chang, C.H.: Prediction-based watermarking schemes using ahead/post AC prediction. Signal Process **8**(90), 2552–2566 (2010)
17. Anderson, R.J., Manifavas, C.: Chameleon, a new kind of stream cipher. In: Proceedings of the 4th International Workshop on Fast Software Encryption, pp.107–113, 1997
18. Lian, S.: Multimedia content encryption: techniques and applications. CRC Press (Auerbach Publications) (2008)
19. Iwakiri, M., Thanh, T.M.: Incomplete cryptography method using invariant Huffman code length to digital rights management. In: The 26th IEEE International Conference on Advanced Information Networking and Applications (AINA-2012) (2012)
20. Thanh, T.M., Iwakiri, M.: A proposal of digital rights management based on incomplete cryptography using invariant Huffman code length feature. J. Multimed Sys, pp. 1–16 (2014) (ISSN 1432-1882)
21. Lee, Y., Park, S., Kim, C., Lee, S.: Temporal feature modulation for video watermarking. IEEE Trans. Circuits Syst. Video Techn. **19**(4), 603–608 (2009)
22. Wang, L., Ling, H., Zou, F., Lu, Z.: Real-time compressed–domain video watermarking resistance to geometric distortions. IEEE MultiMedia **19**(1), 70–79 (2012)
23. 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)
24. Matsui, K.: Fundamentals of digital watermarking. Morikita-publisher (1998) (in Japanese)
25. Test images: Computer Vision Group, University of Granada, Spain, http://decsai.ugr.es/cvg/index2.php. (2008)
26. Thanh, T.M., Hiep, P.T., Tam, T.M., Tanaka, K.: Robust semi-blind video watermarking based on frame-patch matching. Int J Electron Commun AEU 68(10), 1007–1015 (2014) (Elsevier, ISSN: 1434-8411)
27. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process **13**(4), 600–612 (2004)