Fingerprinting Method using Invariant Offset Huffman Code Length Feature

Ta Minh Thanh
Department of Computer Science
Tokyo Institute of Technology
2-12-2, Ookayama, Meguro-ku,
Tokyo, 152-8552, Japan.
Email: thanhtm@ks.cs.titech.ac.jp

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

Abstract—This paper describes an experimental system integration of Digital Rights Management (DRM) architecture and fingerprinting which is composed of an incomplete cryptography scheme using invariant offset huffman code length feature and the user identification mechanism to control the quality of the digital contents. We adopt the AC-coefficient huffman code length feature of the DCT coefficient in the JPEG codec to the implement incomplete cryptography. In the our scheme, the copyright information is embedded into the decoded content while decoding process, and the size of digital contents are invariant in the whole process. Experimental results with simulation confirmed that the modified codes keep compatibility with standard JPEG format, and revealed the proposed method is suitable for DRM in the network distribution system.

I. INTRODUCTION

Digital content is increasingly in digital form and is distributed using the Internet. The ease of copying has created a need to develop a means to protect it. DRM tries to find a solution to this problem inside a triangle set by technology, economics and law.

Normally, conventional DRM system can prevent from thus piracy of the contents as cutting the user off from an illegal approach to the content through the encryption of it [1]. However, conventional DRM technologies are manipulated by encryption and fingerprinting method separately. Therefore, original content is disclosed temporarily inside a system in the user’s decryption (key management process) [2]. In that case, the original content is able to be distributed without any fingerprinting information and distribute via network. And it also can’t trace the illegal distributor. In addition, after embedding, the size of fingerprinted content can not be controlled, then it is various for individual user.

In this paper, we describe design and implementation of DRM technique with fingerprinting based on an incomplete cryptography system by using the invariant offset huffman code length feature. The know-how of the proposed method is the fundamental incomplete cryptography[3]. The incomplete cryptography is proposed for improving the problem of conventional DRM system. Our method will deteriorate the quality of original contents to make trial contents for distribution to wide users via network. The quality of trial contents will be controlled with a watermarked key at the incomplete decoding process, and user information will be embedded into the incomplete decoded contents simultaneously.

The rest of the paper is organized as follows. Section 2 introduces the incomplete cryptography. Section 3 describes the summary of huffman code in JPEG codec[6] and shows the problem of file size after embedding. Section 4 proposes the new DRM system using invariant offset huffman code length feature. Section 5 presents the results of the experiments. Section 6 concludes the paper.

II. OVERVIEW OF INCOMPLETE CRYPTOGRAPHY

The incomplete cryptography[3] consists two steps: the incomplete encoding and the incomplete decoding (see Fig.1).

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

$$C = E(k, P)$$

(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 $k' \neq k$.

$$P' = D'(k', C)$$

(2)

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

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

III. HUFFMAN CODE IN JPEG AND THE PROBLEM OF CONVENTIONAL WATERMARKING METHOD

There are several approaches related to huffman code watermarking [3], [4], [5]. These methods were studied with viewpoints of the watermark robustness. Those were developed to maintain the high quality even after embedded copyright information in the digital content. However, there is a problem in these methods: the watermarked content is changed because the size of the content is determined by the length of codes in the huffman table.

A. Summary of huffman code in JPEG algorithm

JPEG is an image compression algorithm and image file format of international standard, and it is used in global applications now[6].

Images subjected to JPEG encoding are first broken down into $8 \times 8$ 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 ($8 \times 8$) 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 $8 \times 8$ block are often become “0” after quantization, the spectrum of picture tends to be constructed with low frequency coefficients.

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

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

B. Huffman code of DC coefficients

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

1) Encoder algorithm:

Step 1. Obtain the difference $diff$ of the DC coefficients in the previous DCT blocks.

Step 2. If the most significant bit(MSB) of the additional bits is 0, then the $diff$ is negative value. After obtaining $diff + 1$, add bit “1” before least significant bit(LSB) of $(S + 1)^{th}$ bit and change to negative value.

2) Decoder algorithm:

Step 1. Take the huffman code from the extracted JPEG bitstream and find out a category $S$ from Tab.I.

Step 2. From the combination of $R_c$ and $S$ in Tab.I, we can take the variable-length codes for each AC coefficient.

Step 3. Coupling the variable-length codes and the additional bits corresponding to $S$.

C. Huffman code of AC coefficients

Tab.II is used for the AC coefficients in the encoder/decoder as the huffman codes.

1) Encoder algorithm:

Step 1. Obtain the difference $diff$ of the DC coefficients in the previous DCT blocks.

Step 2. Find the category $S$ of $diff$ in Tab.I and take variable-length codes and additional bits corresponding to $S$.

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

2) Decoder algorithm:

Step 1. Take the huffman code from the extracted JPEG bitstream and obtain $R_c$, the nonzero AC coefficient by referring Tab.II.

Step 2. If the most significant bit(MSB) of the additional bits is 0, then the $diff$ is negative value. After obtaining $diff + 1$, add bit “1” before least significant bit(LSB) of $(S + 1)^{th}$ bit and change to negative value.

Step 3. From the combination of $R_c$ and $S$ in Tab.II, it is obtain the huffman code as the AC coefficient.
TABLE I
HUFFMAN CODE TABLE FOR DC COEFFICIENTS.

<table>
<thead>
<tr>
<th>diff</th>
<th>S</th>
<th>Huffman code</th>
<th>Additional bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2047,...-1024,1024,...,2047</td>
<td>11</td>
<td>111111110</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>-1023,...-512,512,...,1024</td>
<td>10</td>
<td>11111110</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>-511,...-256,256,...,511</td>
<td>9</td>
<td>11111111</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>-255,...-128,128,...,255</td>
<td>8</td>
<td>111111</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>-127,...-64,64,...,127</td>
<td>7</td>
<td>111111</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>-63,...-32,32,...,63</td>
<td>6</td>
<td>11111</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>-31,...-16,16,...,31</td>
<td>5</td>
<td>11111</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>-15,...-8,8,...,15</td>
<td>4</td>
<td>11111</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>-7,...-4,4,...,7</td>
<td>3</td>
<td>11111</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>-3,-2,2,3</td>
<td>2</td>
<td>11111</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>-1,1</td>
<td>1</td>
<td>11111</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>11111</td>
<td>000000000000,...,011111111110000000000000000,...,11111111111</td>
</tr>
</tbody>
</table>

TABLE II
HUFFMAN CODE TABLE FOR AC COEFFICIENTS.

<table>
<thead>
<tr>
<th>Rc</th>
<th>S</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1010(EOB)</td>
<td>00</td>
<td>01</td>
<td>...</td>
<td>1111111110000011</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>non</td>
<td>1100</td>
<td>1101</td>
<td>...</td>
<td>1111111101001000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>non</td>
<td>1100</td>
<td>1101</td>
<td>...</td>
<td>1111111101001000</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1111111001(ZRL)</td>
<td>1111111111111101</td>
<td>1111111111111110</td>
<td>...</td>
<td>1111111111111110</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. The problem of watermarking method after embedding.

Step 4. If S is not 0, the additional bits is obtained. If the MSB of the additional bits is 0, the AC coefficient is negative value. After AC +1, add bit “1” before least significant bit (LSB) of (S+1)th bit and change it to negative value. Repeat this step to finish the decode of DCT block.

According to this algorithm, the DC/AC coefficients in JPEG image are encoded and decoded to the huffman code. Therefore, after applying the information embedding, the size of image is changed because the huffman code of image is already changed.

D. The variant file size problem in conventional watermarking method

In general, almost of conventional watermarking method is not considered the huffman code. That is the reason why the size of digital content is changed before and after embedding information into digital content. For showing the problem of conventional watermarking method, we describe an example of paper [3] in Fig.2 which is not considered the huffman code in AC coefficient.

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

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

From Step 1~4, huffman code bits of AC coefficients are created,
\{10110000 10110110 0111 1010\}
and bitstream length is **24 bits**.

To scramble for making trial content C, P is encrypted and result is shown in Fig.2(b). The zigzag scanning result is \{22, 10, 15, 16, EOB\}. As Step 1~4 above, if these AC coefficients in this table are converted to the huffman code, the result is
\{1011010 10111111 1101001000 1010\}
and the bitstream length is **30 bits**.

On the other hand, after the decoding Fig.2(b), Fig.2(c) is obtained and it is also the watermarked DCT table. Again, we observe the zigzag scanning result \{2, 9, −9, 4, EOB\}. We
calculate the huffman code of this DCT table as the Step 1~4 above and obtain the huffman code is,
\[
\{10110001\ 10101110\ 1000100\ 1010\}
\]
and the bitstream length is 27 bits.

According to the results of huffman code in DCT table of original image, scrambled image and watermarked image, we recognize that the length of huffman code is variant through processes (24 bits, 30 bits and 27 bits, respectively) and it is the reason of variant file size because of file size is decided by huffman code length of each DCT table in JPEG image.

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

IV. THE PROPOSED DRM USING INvariant OFFSET HUFFMAN CODE LENGTH JPEG CODEC

Fig.3 describes an proposed idea for design of DRM and fingerprinting system based on incomplete cryptography[3]. Our fingerprinting method uses the invariant offset huffman code length feature of difference AC-category in huffman table (see Tab.II) to implement the DRM system and improve the problem of content size after embedding.

Before distribution, producer \(T\) has a digital content \(P\) (e.g. JPEG image) and needs to be sent to users as much as possible. Thus, \(T\) creates a scrambled content \(C\) with encryption key \(k\) and an encode function \(E\) based on incomplete cryptography. A pair of AC coefficient \(\{p_1^S, p_2^S\}\) of \(P\), which belongs to the same category \(S\) in huffman table, is chosen for encryption. \(E\) is used to encode \(\{p_1^S, p_2^S\}\) by replacing with \(\{p_1^{S+M}, p_2^{S-M}\}\) that belongs to category \(S + M\) and \(S - M\), respectively. \(C\) 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 a trial content.

After trial \(C\), \(R\) has to register his/her individual information. This information will be used as the watermarked information \((W_i)\) and to be embedded into the content. When \(T\) receives the purchaser’s agreement, \(T\) sends a watermarked key \(k\) to \(R\). \(k\) is the incomplete decoding key and it is prepared individually to each user by using \(W_i\) and generated function \(G\). To create the \(k\), \(T\) extracts one bit \(b\) from \(W_i\). If \(b = 0\), \(k\) is used to control the decryption function \(D\) decodes the pair \(\{p_1^{S+M}, p_2^{S-M}\}\) to original value \(\{p_1^S, p_2^S\}\). Otherwise, \(D\) decodes the \(\{p_1^{S+M}, p_2^{S-M}\}\) to \(\{p_1^S, p_2^S\}\) that belongs to the same category \(S\). These decoded positions are recorded into the secret key \(k_s\) that is used for tracing illegal user.

\(R\) decodes \(C\) using \(k\) and \(D\) to obtains the high quality content \(P'\). In this decoding process, ID information \((W_i)\) of user is embedded in \(P'\) as the copyright information.

Therefore, when a producer wishes to check whether the users is a legal user, he/she can extract the watermarking information from \(P'\) and compare with his user database. If the watermarking information matches his database, the user is a legal user. Conversely, if the watermarking information is a 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 informs the producer about the existence of watermarking which can easily identify users, and limit the illegal redistribution in advance.

V. EXPERIMENTAL RESULTS

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

To prove the efficiency of proposed method, we used 256×256, 8-bit images in SIDBA database and created JPEG images with quality 75. We also used PSNR (Peak Signal to Noise Ratio)[3] to evaluate the JPEG image quality. On the other hand, we prepared a bitstream 32 × 32 pixels of binary picture (Nda32) as the watermarking information.

An example of implementation is shown in Fig.4. Firstly, we calculate the huffman code length of original DCT

\(^1\)\url{http://gcc.gnu.org/}
\(^2\)\url{http://www.imagemagick.org/script/}

![Fig. 3. Overview of proposal system.](image-url)

![Fig. 4. Example of AC coefficient implementation.](image-url)
After decoding, we can obtain the high quality watermarked content. The size of \( P, C \) and \( P' \) is not changed for the whole encoding/decoding process. Otherwise, we can extract \( W_i \) (Fig.5(d)) from \( P' \) and we can compare with original user information to confirm the legal user.

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

### VI. Conclusion

In this paper, we have presented a scheme of an incomplete cryptography system and proposed the digital content distribution system based on incomplete cryptography using invariant offset huffman code length feature. According to the above results, we conclude that the above technical problem by the conventional DRM system is solved by using the incomplete cryptography system and we can control the content size without changing the whole system.

### REFERENCES