SHORT N-SECURE FINGERPRINTING CODE FOR IMAGE

Won-gyum Kim, Youngho Suh

Digital Contents Research Division, Electronics & Telecommunications Research Institute (ETRI)
161 Gajeong-dong, Yuseong-gu, Daeduk Science Town, Daejon, Korea
E-mail: {wgkim, shy}@etri.re.kr

ABSTRACT

Fingerprinting is one of many copyright protection methods and an extended technique of watermarking. In fingerprinting the identity of customer is embedded into the content in a way that is difficult to erase. The main difference between watermarking and fingerprinting is that different copies for each customer can be produced. Attackers compare several fingerprinting copies and find the location of the embedded information and destroy it by altering the values in those places where a difference was detected. In this paper, we propose a short fingerprinting code which is robust to the collusion attack of *N* customers. In the proposed fingerprinting scheme, two kinds of code are used. The first code has the same value for all customers and is embedded at the same location like watermark. The second code is designed unique for each customer. We apply this scheme to the digital image and show that it has some robustness against collusion attacks.

1. INTRODUCTION

Digital watermarking is a technique to embed extra information imperceptibly into digital contents such as a still image, an audio in audio track or a movie. In watermarking the same information is embedded into all copies of contents because it identifies the ownership of the contents. Another application of digital watermarking is fingerprinting. The fingerprinting embeds a unique customer identification as a watermark into the content. This will enable the owner (or distributor) to trace an unauthorized copy back to the source.

There is important difference between watermarking and fingerprinting. In watermarking the mark is the same for all customers but the mark is not the same in fingerprinting. It depends on the customer's identity (or serial number of contents). The owner can trace the origin of illegal copy by extracting the customer's identity.

But we have to consider another problem because of this difference. Attackers can use this to remove fingerprinting code by comparing their copies and try to locate and delete some marks. It is known as collusion attack. An attacker colludes with several fingerprinted contents and finds differences and tries to remove fingerprinting information. One of the simplest approaches to performing a collusion attack is to average multiple copies of the content together[1]. Other collusion attacks might involve forming a new content by selecting different pixels or blocks from the different colluders' content. By gathering a large enough coalition of colluders, it is possible to sufficiently attenuate each of the colluders' fingerprints and produce a new version of the content with no detectable fingerprints. It is therefore so important to design a fingerprinting code that resists collusion that we discourage attempts of collusion by the customers. Many collusion secure fingerprinting codes have been proposed[2][4][5][6], but they still have a big problem in that the length of the code is increased geometrically by the number of customers.

In this paper, we propose a short fingerprinting code which resists *N* colluders and show experimental results with digital images. In Section 2, we describe some methods of collusion attack and fingerprinting. Section 3 shows how to construct n-secure fingerprinting code and the experimental results and conclusion are shown in section 4 and 5.

2. COLLUSION AND RELATED WORKS

In this section we describe some kinds of collusion attacks and discuss existing fingerprinting codes. Wahadaniah introduced four kinds of collusion attacks including the averaging attack[1]; Averaging, Maximum-minimum, Negative and zero correlation and New-zero correlation attack. In the case of new zero-correlation collusion attack only 3 contents are enough to remove fingerprinting code

0-7803-8554-3/04/$20.00 ©2004 IEEE.

completely if noise-like signal is used as a fingerprint and fingerprints are extracted by correlation detection method.

The early try of collusion secure fingerprinting was evaluated by Dittmann[4]. Dittmann produced collusion secure fingerprinting codes and embedded into the DCT domain of a digital image using a general watermarking mechanism.

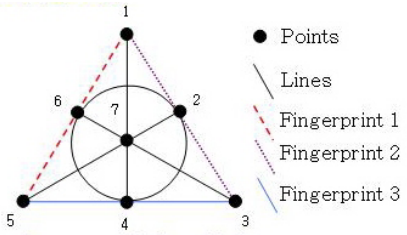


Fig. 1 Finite projective space for 3 customers

To generate collusion secure code a mathematical model called the finite projective space was used. The finite projective space for 3 customers is shown in fig. 1. For example, three fingerprinting codes are produced from fig.1 and these codes are as follows:

Customer 1 : 1 0 0 0 1 1 0
Customer 2 : 1 1 1 0 0 0 0
Customer 3 : 0 0 1 1 1 0 0

Main property of this code is that there is a common bit position in any pair of codes. The first bit between customer 1and 2, and the third bit between customer 2 and 3 are common bits. In other words the location of common bit presents the customer identity because it is not destroyed after collusion attack. In this model the number of colluders to be allowed is very small and the length of the code is increased at an exponential rate in proportion to the number of customers.

The other approach was performed by Sebe[6]. Sebe proposed 3-secure fingerprinting codes whose length is shorter than Boneh-Shaw's[3]. Sebe used a dual binary Hamming code to construct collusion-secure fingerprinting codes. The dual code of a binary Hamming code is a binary code with 2^n codewords of length $N=2^n-1$ such that the distance between any two codewords is 2^{n-1}. In dual binary Hamming code proposed by Sebe colluders are detected by using the property that the hamming distance between the new collusion code and the fingerprinting code joining the collusion becomes shorter.

The disadvantage of this code is that, in the exceptional case, the fingerprinting code which does not join the collusion might be included. To solve this problem Sebe proposed a scattering code, which is an expanded version of the dual Hamming code. But, in practice, it is not possible to embed the code into the content which has a limited size, because the length of the scattering code is still very long.

The common property of the collusion secure fingerprinting codes proposed before is that the common bits of the any pair of codes always exist in the code and they are unique, but the length of the code is still long. In this paper, we propose a collusion secure fingerprinting code for N customers shorter than Dittmann's and dual binary Hamming code.

3. N-SECURE FINGERPRINTING CODE

Before describing the proposed scheme, we assume two conditions. These conditions will be connected to the experimental results. The first condition is that colluders can not change the state of an undetected code without rendering the content useless. It is the Marking Assumption described by Boneh-Shaw[3]. The second one is that colluder can remove the detected code perfectly. In other word the collusion attacks we introduce in section 2 are strong enough to destroy the detected fingerprinting code.

The fingerprinting code we proposed is designed to minimize the code length. The length of the proposed code is increased in proportion to the number of customers. In the general fingerprinting scheme we already describe above all fingerprinting code are destroyed by collusion attack if the code value is not the same, so the same code value is embedded in the specific location. That is the reason why the length of code has to be long. The basic idea of the proposed code is that the location of the destroyed code after collusion attack is also used to identify a customer.

Two codes are used in the proposed fingerprinting code. The value of the fist code is '1' and is embedded into the same location of the image like a watermark. This code determines if the fingerprinting code exists in the image or not and can't be removed by the collusion attack because the value and the location are same. The second is a code for identifying customers. This code is designed to be different and unique depending on the customers. The formula of the n-secure fingerprinting code for customer N is as follows:

$$C_i = \{1, C_1, C_2, ..., C_n\}, \quad i = customer \quad index$$

$$C_n = \begin{cases} 0, & if\ (n = customer \quad index) \\ 1, & Otherwise \end{cases}$$

We can say the code in the first location is a kind of signal watermark. Only if the signal watermark is detected, then the fingerprinting code is searched. Bit '0' means that no

fingerprinting code is embedded in this position. For example, we show 8-secure fingerprinting codes as below:

customer 1 (C_1) : 1 0 1 1 1 1 1 1 1
customer 2 (C_2) : 1 1 0 1 1 1 1 1 1
customer 3 (C_3) : 1 1 1 0 1 1 1 1 1
customer 4 (C_4) : 1 1 1 1 0 1 1 1 1
customer 5 (C_5) : 1 1 1 1 1 0 1 1 1
customer 6 (C_6) : 1 1 1 1 1 1 0 1 1
customer 7 (C_7) : 1 1 1 1 1 1 1 0 1
customer 8 (C_8) : 1 1 1 1 1 1 1 1 0

The length of code C_i is $N+1$. The extracted code when customers 1 and 2 collude with is {1, X, X, 1, 1, 1, 1, 1, 1}. X means that 'not detectable'. The code information where the values are different, i.e. the positions 1 and 2, is destroyed and the remaining code is a unique code sequence. So we can predict that customers 1 and 2 joined to the collusion. For another example, suppose that the case of joining to the collusion for customer 4 through 8. The detected code sequence is {1, 1, 1, 1, X, X, X, X, X}. We notice that customer 4, 5, 6, 7, and 8 are members of collusion attack by the position of X. In the worst case, suppose that all customers join to the collusion. The extracted code is {1, X, X, X, X, X, X, X, X}. Every code is not detected except the signal mark located at the first position. In this case we can predict all customers take part in the collusion attack.

4. EMBEDDING AND EXTRACTION

In this section we describe how to embed and extract fingerprinting code in a digital image. General image watermarking technique is used to embed the fingerprinting code. The watermarking algorithm we use is designed not to use the original image in the retrieval process. We also do not consider RST attack.

4.1 Embedding

The detailed embedding of the fingerprinting code for each customer's content is performed in three steps. In the first step the fingerprinting code for the customer is generated. The number of customers who can be delivered depends on the size of image and the redundancy for watermarking strength. In step 2 a random sequence, consisting of {1, -1}, is generated from the user secret key. The length of the sequence is the same as the block size. In step 3 the image is blocked and the random sequence is embedded according to the fingerprinting code. If fingerprinting code is 1, then the random sequence is embedded in the block.

Algorithm 1 (code embedding)

1. Parameters
- *Image I[i, j] with height h and width w*
- *Binary fingerprinting code with length N+1 : FC[n]*
- *Secret user key : k*
- *Random sequence with length L = (Image size)/N+1 : W[i, j]*
- *Adaptive factor : $\alpha[i, j]$*

2. Chose fingerprinting code for customer
3. Calculate N+1 blocks of the image
4. Calculate adaptive factor from the masking model
5. For k=0 to N+1 do:
if FC[k]='1'
*Add $\alpha[i, j]$ *W[i, j] to I[i, j];*
else
Nothing

In this paper, we embed fingerprinting code into the spatial domain of the image. The test image is 512x512 gray-scaled 'Lena' image. For convenience the image is divided into 16 small blocks($N+1=16$) without any security condition. Therefore the number of customers who can be delivered is 15. In practice, the embedding algorithm is designed with specific marking positions in each copy of the image.

To increase robustness and imperceptibility we decide α value using a simple adaptive masking model. In this model two factors are used; a local variance and a maximum local variance. We calculate the scaling factors of these two variances through simulations. Fig. 2 shows the result of the masking model we used in this experiment. Due to the adaptive factor the fingerprinted image has no visible artifacts with PSNR 40.2dB for Lena.

Fig. 2 An adaptive masking model of the Lena image

The embedding process we used in this paper is simple and fast because it is done in the spatial domain of the image. To have robustness to other attacks like compression, filtering and etc., it is very important to choose the proper size of the block. But we do not consider these attacks in this experiment.

4.2 Extracting

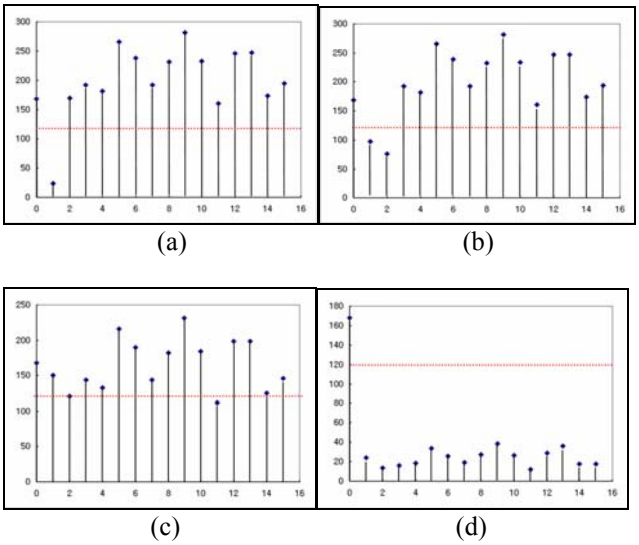
In the detection process we calculate cross-correlation values between the random sequence generated by user secret key, k and coefficient value of the fingerprinted image.

$$Corr[n] = \sum (HPF(X'_{[i,j]}[n]) * W_{[i,j]}[n])$$

W is a reference watermark generated by the user secret key, *X'* is a coefficient value of the fingerprinted image and *n* is a block index. HPF means a high-pass filtering to reduce the probability of the false positive alarm because the watermark is a noise-like signal and mainly embedded in the edge area of the image. *Corr* is a correlation table of the image block. If the correlation value of a block is over threshold value, we retrieve a '1' and a *X* otherwise.

4.3 Collusion

Three collusion examples and detection statistics are shown in Figure 3. We simulated two kinds of collusion attacks: averaging and zero-correlation. In fig. 3 (a) shows correlation values for user code 1 without collusion attack. We notice that the correlation value of user 1 only is small. In the case of (b) users 1 and 2 performed averaging, result in the output of the detector as {1, X, X, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}.



(a) (b)

(c) (d)

Fig.3. Detection values for 2 users' and 15 users' collusion. (a) user code 1 case without any collusion, (b) collude users 1 and 2 with averaging attack, (c) collude all 15 users with averaging attack, (d) collude all users with zero-correlation attack. The dotted line is threshold.

The case of (c) is detection values after performing averaging attack for all 15 customers. In proposed scheme averaging is not a cost-effective attack because the fingerprinting code of only one user is different. This means the energy of watermark is increased after collusion attack. In the case of (d) we performed zero-correlation collusion attack for all 15 users. Only the first fingerprinting code was detected because his attack removed fingerprinting code of all users perfectly

5. CONCLUSION

In this paper, we proposed a short n-secure fingerprinting code and apply to the digital image and show that it is practical because the length of the code is shorter than the length of the proposed ones before. The fingerprinting code we proposed is designed to minimize the code length. The length of the proposed code is increased in proportion to the number of customers. In the general fingerprinting scheme all fingerprinting codes are destroyed after collusion attack if the code value is not the same, so the same code value is embedded in the specific and unique location. The basic idea of the proposed code is that the location of the destroyed code after collusion attack is also used to identify a customer. Our codes are efficient in that they require only *N+1* bits to accommodate *N* customers.

6. REFERENCES

[1] V. Wahadaniah, Y. L. Guan, and H. C. Chua, "A New Collusion Attack and Its Performance Evaluation," *Proceedings of IWDW2002*, pp.88-103, 2002

[2] J. Kilian, F. T. Leighton, L. R. Matheson, T. G. Shamoon, R. E. Tarjan and F. Zane, "Resistance of digital watermarks to collusive attacks," *Proc. of IEEE International Symposium on Information Theory*, pp.271, 1998

[3] D. Boneh, J. Shaw, "Collusion-Secure Fingerprinting for Digital Data," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1897-1905, Sept. 1998

[4] J. Dittmann, "Combining digital watermarks and collusion secure fingerprints for customer copy monitoring," *Proc. IEE Seminar Sec. Image & Image Auth.,* pp.128-132, March 2000

[5] J. Domingo-Ferrer and J. Herrera-Joancomartí, "Simple Collusion-secure Fingerprinting Schemes for images," in *IEEE International Conference on Information Technology: Coding and Computing, ITCC'2000,* pp. 128-132. ISBN 0-7695-0540-6

[6] F. Sebe and J. Domingo-Ferrer, "Short 3-Secure Fingerprinting Codes for Copyright Protection," *Lecture Notes in Computer Science*, Vol. 2384, pp.316-327, 2002