

Hybrid Approach for Image Protected Shares Based on Visual Cryptography and Fragile Watermarking Scheme

Surendra K. Raut¹, Durgesh Singh², Shivendra Shivani³ and Suneeta Agarwal⁴

^{1,3,4}*Department of Computer Science and Engineering, MNNIT Allahabad
Allahabad-211004, India.*

²*Department of Computer Engineering, IIT (B.H.U.) Varanasi,
Varanasi-221005, India.*

Abstract

In Visual Cryptography (VC), shares are the most sensible objects and may be tampered by any unauthorized person. So the protection of VC shares is most essential. This paper proposes a hybrid approach in which shares of VC are protected through the self embedding fragile watermarking technique. At the receiver end, these protected shares are checked for any alteration by recalculating watermark and comparing with the existing watermark in protected shares. If any mismatch is found, altered pixels are marked so that receiver can request the sender to retransmit the particular shares, otherwise shares will be stacked together after discarding the watermark and recover the secret image.

Keywords: Visual Cryptography, Fragile Watermarking, Self Embedding, Protected Shares.

1. Introduction

In 1994, Naor and Shamir [1] proposed a new cryptography area, Visual Cryptography (VC). It is a secret sharing technique used for sharing of visible object (i.e. text, image etc.) secretly. The most notable feature of this approach is that it allows the retrieval of the secret information without any cryptographic computation because decryption is done by human visual system only. In this technique secret information is distributed in the form of shares to different users. Each share shows nothing more than a random binary pattern which does not reveal any information about the secret. The secret

information can be recovered only when all the distributed shares are collected together and stacked on top of one another.

Digital watermarking is a process for inserting watermark (ownership information or hidden information or mark) into cover signal such as audio or image data. At any given moment, the hidden information can be extracted to prove ownership or to ensure integrity or simply to get some copyright-related information. Image, text, video or any logo which can be notified as courtroom evidence can be the ownership information. Digital watermarking technique can be categorized into three categories namely robust, fragile and semi fragile watermarking [3] [4]. In robust digital watermarking applications, the watermark is extracted even if modification is strong. However, extraction is failed in case of fragile watermarking. Fragile watermarking has come into picture for ensuring the legitimacy and data integrity [2][3]. Fragile watermarking can be achieved block-wise or pixel-wise depending upon insertion of the hidden information to the cover image.

The remaining paper is organized as follows: the proposed scheme is given in section 2 and in section 3 the experimental results are shown. The paper is concluded in section 4 followed by the references.

2. Proposed Algorithm

The proposed approach has two phases. First one is encryption and second one is decryption. In the encryption, input secret image is encrypted into two protected shares based on self embedding fragile watermarking scheme while in decryption individual share is checked for any alteration. If any possible alteration (i.e. modification) is detected to the share, then the alteration is marked and shown as tampered share and receiver requests the sender to resend new genuine share, otherwise two shares are stacked together after discarding watermark information to get the secret image. Here Figure 1 and Figure 2 show the encryption procedure and the decryption procedure respectively.

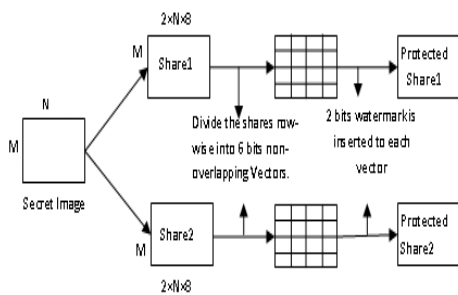


Figure 1: Block diagram for encryption procedure.

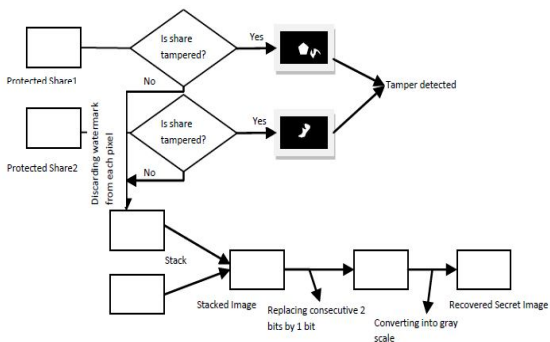


Figure 2: Block diagram for decryption procedure.

2.1 Encryption Procedure

The encryption procedure consists of following steps:

- Step-1:** The gray value of each pixel of input image is converted into eight bit binary representation.
- Step-2:** Do two bits pixel expansion for each bit to generate shares using table 1. For example if the bit value is 1 in the image then, either (0, 1), (1, 0) or (1, 0), (0, 1) will be stored in share1 and share2 respectively.

Table 1: Pixel Expansion encoded.

	For bit values 1		For bit values 0	
Share1	0 1	1 0	0 1	1 0
Share2	1 0	0 1	0 1	1 0
After stack	1 1	1 1	0 1	1 0

- Step-3:** To generate protected shares, divide each share row wise into six binary bits non-overlapping vectors. Now calculate the two binary bits watermark information for each vector. These two bits generation procedures are as follows.

2.2 Bits Generation

Let the total number of 6-bits vectors in each share be S. Individual binary bit of a vector B_p of share can be written as b_i where $i \in (0 \dots 5)$ and $p \in (1 \dots S)$, the position value of that vector i.e. the row value and column value in which the vector lies are denoted by G_p^r and G_p^c and the binary representation of G_p^r and G_p^c are $b_7^r b_6^r b_5^r \dots b_0^r$ and $b_7^c b_6^c b_5^c \dots b_0^c$ respectively.

Watermark bit₁ (T_{b1}) Generation:

Generate the Watermark bit₁ in following manner:

$$X = Ex - OR(b_i^r, b_i), i = 5,4 \dots 0 \tag{1}$$

$$Y = Ex - OR(b_i^c, b_i), i = 5,4 \dots 0 \tag{2}$$

Where, X represents bitwise Ex-OR operation between 6 binary bits of each vector and 6 LSB of row value of the corresponding vector. Similarly Y represents bitwise Ex-OR operation between 6 binary bits of each vector and 6 LSB of column value of corresponding vector.

$$T_{b1} = \left(\sum_{i=1}^6 (X_i \times Y_i) \right) \text{mod } 2 \tag{3}$$

Watermark bit₂ (T_{b2}) Generation:

This bit for each vector can be calculated from six binary bits of that vector itself. First we will make the pair of each two consecutive bits and take the Ex-OR operation of bits of each pair, so there will be five bits, then add all the five bits and take mod 2 operation of the sum of the five bits.

$$T_{b2} = \left(\sum_{i=0}^4 (b_i \oplus b_{i+1}) \right) \text{mod } 2 \quad (4)$$

After calculating two bits watermark for each vector of the shares, convert these eight bits (i.e. six binary bits of each vector and two bits corresponding watermark, T_{b1} as first LSB and T_{b2} as second LSB) into single gray scale pixel for each vector and called them as protected shares. Finally these protected shares will be transmitted to the receivers.

2.3 Decryption Procedure

Protected share1 and protected share2 are first checked for alteration, If any alteration is made to the shares then the alter portion is marked and shown as tampered one, otherwise shares are stacked and the original secret image is recovered. This decryption procedure can be done in following:

- Step-1:** Using six MSBs of each pixel of protected shares recalculated T_{b1} and T_{b2} using equation 3 and equation 4, and match with first and second LSB of that pixel. If any mismatch is found then that pixel treated as altered one otherwise treated as unaltered.
- Step-2:** If the protected share1 and/or protected share2 is/are altered, then the portions where alteration (corresponding pixel) are marked and shown as tampered share so that receiver can request sender to resend new genuine share.
- Step-3:** If the Protected share1 and protected share2 are unaltered, then these two shares are bitwise stacked after removing watermark data and resulting image known as stack image. This stack image is binary image.
- Step-4:** Replace the consecutive two bits of the stack image by one bit. From the Table 1, it is clear that for the bit value 1 in the original secret image the corresponding two bits in stack image are same i.e. (1,1) and that for bit value 0 are different i.e. (1,0) or (0,1). So the algorithm scans entire stack image and replace each two consecutive bits by one bit (like set the bit value as 0, if two consecutive bits differ otherwise set bit value as 1 in the stack image). The resulting image will be of size of $M \times 8N$ and convert this into the corresponding gray scale image and get the original secret image.

3. Experimental Results

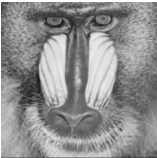
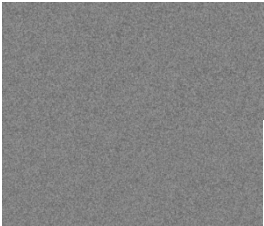
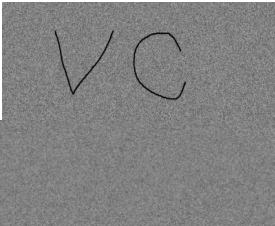
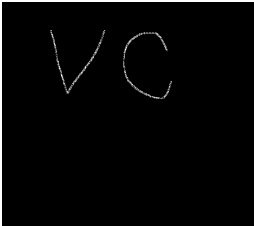
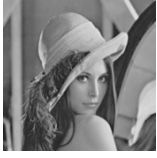

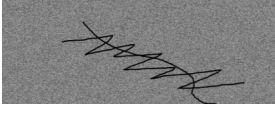

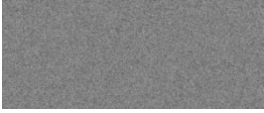
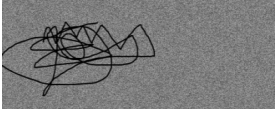

To show the efficiency and accuracy of this approach, we demonstrate it with the help of 4 examples as shown in Figure. 3. Images of size 256×256 are taken from the standard image database like Baboon, Lena, Vegetable and Camera man. For each image, two protected shares, protected share1 and protected share2 are generated. In the tamper detection column of Figure 3, white region shows altered pixel where as black region shows unaltered one.

In the first three examples (a, b, c) shares are tampered and this is accurately detected by this algorithm (white pixels in tamper detection column in Figure 3 shows the tamper area). In last example (d), no share is tampered so there is no white pixel in tamper detection column, these shares can stack together to recover secret image.

The observation during experimental results is shown in Table 2. The detection rate is very high which shows the efficiency of our proposed approach.

Table 2: Essential information observed during encryption and decryption of VC.

Secret Image	No. of altered pixels	No. of detected pixels	Detection Accuracy (%)
Baboon	131	119	96.94
Lena	595	586	98.48
Vegetable	2689	2635	97.99

Original Secret Image	Protected Shares	Tamper Shares	Tamper Detection
 a.			
 b.			
			
Camera man	0	0	100

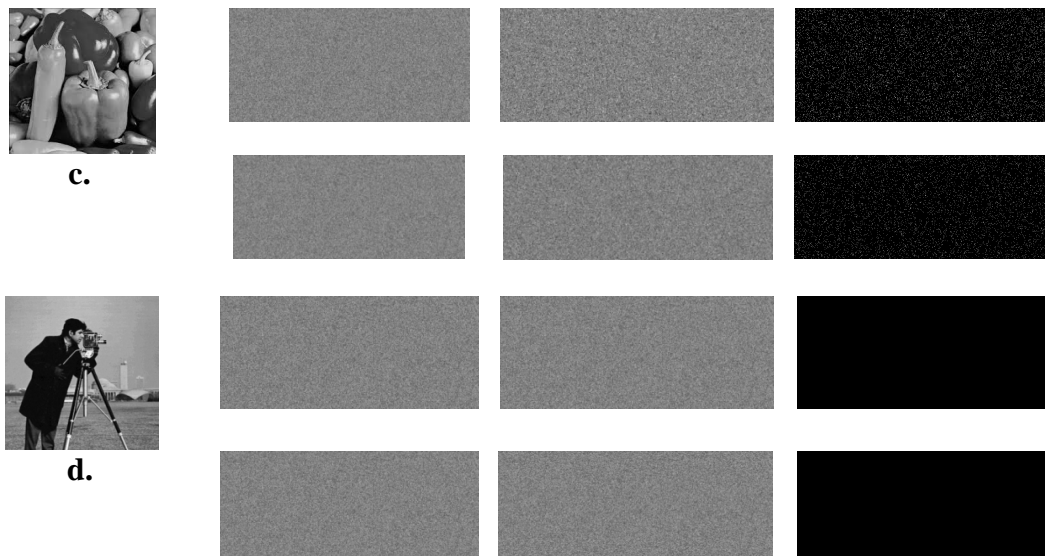


Figure 3: a. Baboon b. Lena c. Vegetable d. Camera man

4. Conclusions and Future Works

This paper suggests an efficient approach which ensures the integrity of the shares made by Visual Cryptography using self embedding fragile watermarking scheme. At the receiver end, these protected shares are checked for any possible alteration by comparing the recalculated the watermark bits with the extracted two LSBs of corresponding pixel of tamper shares. If any mismatch found it shows that the pixel has been altered, altered pixels are marked and shown so that receiver can request the sender to retransmit shares, otherwise shares will be stacked together to recover the secret image. The rate of detection of altered pixels is very high. In future recovery of tamper shares can be done by adding some new features.

References

- [1] M. Naor, A. Shamir, Visual cryptography, Advances in Cryptology Eurocrypt '94, Lecture Notes in Computer Science, Springer, Berlin, , Vol. 950, pp. 1–12, 1995
- [2] D. Singh, S. Shivani, S. Agarwal, “Self-embedding Pixel wise Fragile Watermarking Scheme for Image Authentication” in International Conference on Intelligent Interactive Technologies and Multimedia, Allahabad(IITM 2013), pp. 111-122, vol. 276, Springer CCIS, ISBN: 978-3-642-37463-0, 2013

- [3] S. Shivendra, A. K. Patel, S. Kamble, S. Agarwal, "An effective pixel-wise fragile watermarking scheme based on ARA bits." In Proceedings of the 2011 International Conference on Communication, Computing & Security, pp. 221-226. ACM, 2011.
- [4] Zhou, H. Li, P. Yu. "Semi-fragile watermarking technique for image tamper localization." Measuring Technology and Mechatronics Automation, 2009. ICMTMA'09. International Conference on. Vol. 1. IEEE, 2009.

