Safe Secret Image Sharing with Fault Tolerance Key

Wen-Pinn Fang

Yuanpei University, Taiwan, R.O.C.

Summary
This study investigates the security of shared secret images and designs novel methods for polynomial secret image sharing to improve the security of image sharing. Conventional security systems for sharing secret images assume that all pixels are independent. However, this assumption may not always be true. Combining with cryptography methods, the proposed cryptographic method of secret image sharing has three advantages: small share size, good security, and fault tolerance. The proposed method provides more security compared to traditional secret image sharing.

Key words:
Secret image sharing; fault-tolerance; fake contour; Rijndael

1. Introduction

The current era of instant communication requires new methods of protecting, storing and transmitting important data and involves issues such as encryption, digital signature, watermarking, data hiding and secret sharing. Shamir[3] presented the first secret sharing method in 1979. Secret sharing involves transmitting different shares in different channels. Nobody can see the entire secret message with a single share. The only way to obtain a secret is to collect a number of shares that exceeds a predefined threshold. Thus, secret sharing is "fault-tolerant" because if a channel of share is unavailable, then any other channel can be adopted instead. Figure 1 shows the algorithm. Thien and Lin [4] proposed secret image sharing. Their approach was to divide an image into non-overlapping segments and generate shares that are noise-like images. Figure 1 shows that the coefficients of polynomials, which are random numbers in the original secret sharing method, change to pixel values of the secret image. Their method has been shown to be safe in the report. Compare with the most popular cryptographic technique which is encrypting a document with a key. For example, DES[1] and AES[2] are private-key systems both of which employ block-cipher methods that encode and decode messages quickly and with identical keys. Conversely, RSA is another example of a public key system, which is suitable for transmitting small amounts of important data in an open channel. All of these methods are extensively adopted. However, although these methods are useful for transmitting and storing data securely, the data can not be accessed if the channel is unstable or it the key is lost. The intuitive solution is to generate a duplicate key. However, pure duplication increases the risk of the key being stolen. Because of its fault tolerance property, secret image sharing is a better solution. Fig 2. is an example of secret image sharing.

Initialization Phase

1. D choose w distinct, non-zero elements of Z_p, denoted $x_i, 1 \le i \le n$ (this is where we require $p \ge n+1$). For $1 \le i \le n$, D gives the value x_i to P_i. The values x_i are public.

Share Distribution

2. Suppose D wants to share a key $K \in Z_p$. D secretly chooses (indepently at random) t-1 elements of Z_p, $a_1, \cdots a_{t-1}$.
3. For $1 \le i \le n$, D computes $y_i=a(x_i)$, where

$$a(x) = K + \sum_{j=1}^{r-1} a_j x^j \mod p.$$

4. For $1 \le i \le n$, D gives the share x_i to P_i

Fig.1 The Shamir (r,n)-threshold scheme in Zp

Accordingly, advances in secret image sharing imply increasingly complex methods of generating shares. The design and effective management of multiple shares have received considerable attention. [5,6,7,8] The most common share management method is data hiding. These approaches can manage meaningful images more easily than noisy images can. A well-known approach developed by Thien and Lin[5] is secret image with friendly shadow , which generates a meaningful share in one step. Another method of managing shares is to design a special share. For instance, Fang[6] presented a universal share method. The special share recovers different secret images with different groups of dealers. Hence, Chen and Lin[7] presented a progressive image sharing method with spatial domain handling. Fang[8] also achieved the same result in the frequency domain.

Although many studies have discussed secret image sharing, a continuing problem is how to prevent users from viewing the contour of a secret image. The method proposed by Thein and Lin [4] includes a step that permutes the pixels of a secret image before sharing it. Although this step prevents users from seeing a secret

Manuscript received September 5, 2011
Manuscript revised September 20, 2011

image with the naked eye, it may leak information about some nature images. For example, Fang[9] presented a method of recovering a secret image even when the number of shares is insufficient (discussed in the next section). Therefore, this study investigates the safety of traditional secret image sharing and proposes a way to prevent recovery of a secret image when share number is lower than a given threshold.

The rest of this paper is organized as follows Section 2 describes the proposed method. Experimental results are given in Section 3. Conclusions are finally drawn in Section 4 along with recommendations for future research.

Fig.2. The result of the Thein and Lin method without pixel permutation

2. The proposed method

The proposed method for enhancing the security of traditional secret image sharing has two parts, i.e. encoding and decoding. Some terms are defined below before describing the method in detail.

First, all operations are assumed to be in the finite field (Galois Field), denoted as $GF(N)$. Thus, all values are in the range [0, N]. In this study, $N=2^8=256$, since pixel depth is one byte. Notably, this method differs from traditional secret image sharing, which binds all values by a modular prime number, generally 251.

The proposed method attempts to prevent dealers from knowing the position of any pixel. A dealer who does not know the positions of pixels cannot devise a relationship rule to guess the original pixel value when the number of shares is less than the threshold. This objective is achieved in four phases. First, the whole secret image with a key is encoded using a Rijndael-like encryption system. Second, the secret image sharing method of Thein and Lin [4] is adjusted from prime 251 to GF(256) to achieve lossless recovery. Third, the share key is assigned to temporary keys by the Shamir secret sharing method [1]. Fourth, the temporary keys and image shares are combined to create a full set of shares. The algorithm is given below.

Notation:

W: width
H: height
n: number of shares
r: threshold of secret image sharing
K: encoding key
P: secret image
P': image after preprocessing
S_i : i^{th} share
B_i: i^{th} sector
k_i: key of i^{th} share
+: cascade operation
<<: bit-wise operator shift bit
(x,y): position
<<: bit-wise operator shift bit
(x,y): position

Part 1. Encoding
Input : Secret Image ($W\times H$)
Output: n Shares $(W\times H)/r$
Phase 1. Key handling
Step 1 $K=\sum_{i=0}^{32}(P(i,0)<<(i\times 8))$
Step 2 Share Key
Step 2.1 Generate $r-1$ random numbers.
Step 2.2 Adjust the Shamir secret sharing method as in Fig. 1 to share K to key shares, given by k_i.
Phase 2. Preprocessing
Step 1 Divide secret image into non-overlapped sectors, given by B_i, with block size 32 pixels (256 bits).
Step 2 Perform Steps 2.1 to 2.6
Step 2.1 AddRoundKey
Step 2.1 SubBytes
Step 2.2 ShiftRows
Step 2.3 MixColumns
Step.2.4 AddRoundKey
Repeat 2.1 to 2.4
Step 3 SubBytes
Step 4 ShiftRows
Step 5 AddRoundKey
Step 6. Combine all blocks, that is, $P'=B_0B_1B_2\ldots\ldots$
Phase 3. Share generation
Step 1 Divide P' into non-overlapping blocks, with block size of r pixels, given by $\{a_0,a_1,a_2,\ldots,a_r\}$
Step 2 Generate a function $a(x)=\sum_{j=0}^{r-1}a_j x^j$ in GF(256).
Step 3 $s_j(m,n)=a(j)$
Step 4 Combine key and temporary shares, that is $S_i=k_i++s_i$.

The following terms defined in [10] are adopted:

SubBytes—non-linear substitution of each byte with another byte based on a lookup table.

ShiftRows—a transposition step in which each row of the state is shifted cyclically by a certain number of steps.

MixColumns—a mixing operation that operates on the columns of the state, combining the four bytes in each column.

AddRoundKey—each byte of the state is combined with the round key, which is calculated from the cipher key using a key schedule.

The procedures AddRoundKey, SubBytes, ShiftRows and MixColumns described above are the same as those in the Rijndael method, which are omitted here.

Part 2. Decoding

This step, which reverses the encoding method, is briefly described below.

Input : r shares

Output: Secret Image

After collecting any r shares:

Phase 1: Retrieve temporary keys;

Step 1: $k_i=s(x,0)$ $x=[0,r]$

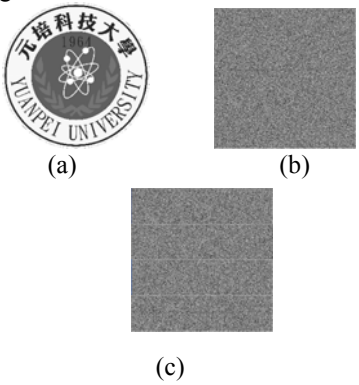
Step 2: compute K with Lagrange interpolation;

Phase 2: determine I' by Lagange interpolation;

Phase 3. Reverse Rijndael to Get P

3. Experiment

According to Fig. 3, the visual effect of the proposed method has a visual effect closely resembles that of traditional secret image sharing with permutation. However, careful analysis demonstrates that a user can determine the relationship among pixels but cannot obtain the secret image without sufficient shares.



(a) (b)

(c)

Fig.3 Experiment result (a) is the original image (b) is image after encryption (c) is shares.

4. Conclusion and Remark

This work presents a novel secret image sharing method. The proposed method has the properties of traditional secret image sharing (including sharing with small size and fault-tolerance) as well as lossless recovery and increased safety. The proposed method is safer than conventional secret image sharing, which does not consider the relationship among neighboring pixels, and requires each dealer to keep a coefficient in the share process. In traditional secret image sharing, the relationship between neighboring pixels is obvious for some images. For example, the relationships in comic-style images or nature images are easily guessed. The proposed method solves this problem by preventing dealers from keeping coefficients in the sharing process and by preventing dealers from knowing the position of each pixel before collecting enough shares.

Since advanced encryption standard (AES) is a fast-block cipher method, the order of complexity of the proposed system is not bigger than that of traditional secret image sharing. In summary, the proposed method is more secure than traditional secret image sharing. Table 1 compares the current and proposed secret image sharing methods.

Table 1. Comparison of traditional secret image and the proposed method

properties	Traditional secret image sharing	The proposed method
Fault tolerance	yes	Yes
Small size share	yes	Yes
Need to keep coefficient	Yes	no
guess relationship	easy	Difficult
Key protection	no	yes

Acknowledgement

The work was supported by NSC project NSC 100-2221-E-264-011-.

References

[1] Douglas R. Stinson, Cryptography-Theory and Practice, Stinson,1995.

[2] John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner, and Doug Whiting, Improved Cryptanalysis of Rijndael, Fast Software Encryption, pp.213–230, 2000.

[3] A. Shamir, “How to share a secret,” Communication of the ACM, Vol. 22, no. 11, pp. 612-613, 1979.

[4] C.C. Thein and J.C. Lin,” Secret Image Sharing”, Computers & Graphics, Vol.26, pp. 765-770, 2002.

[5] C.C. Thein and J. C. Lin, "An Image-Sharing Method with User-Friendly Shadow Images, "IEEE- Transaction on Circuit and Systems, Video Technology, Vol. 13, no.12, pp.1161-1169, 2003.
[6] W.P. Fang and J.C. Lin, "Universal Share for the Sharing of Multiple Images", Journal of the Chinese Institute of Engineers , Vol. 30, no. 4, pp. 753-757, 2007.
[7] S.K. Chen and J.C. Lin, "Fault-tolerant and progressive transmission of images," Pattern Recognition, Vol. 38, pp. 2466-2471, 2005.
[8] W. P. Fang, "Quality Controllable Progressive Secret Image Sharing – Discrete Cosine Transform Approach, "International Journal of Education and Information Technology, Vol.1, pp. 43-47, 2007.
[9] W. P. Fang, "Secret Image Sharing Safety", Proceeding on IEEE International Conference on the 14th Asia-Pacific Conference on Communications (APCC2008), Akihabara, Tokyo, Japan, 2008, 10, 14－2008, 10, 16.
[10] Wikipedia, http://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

Wen-Pinn Fang received his BS degree in mechanical engineering in 1994 from National Sun-Yet-Sen University and his MS degree in mechanical engineering in 1998 from National Chiao Tung University, where he get his PhD degree in Computer Science in 2006 from National Chiao Tung University. His recent research interests include image sharing, pattern recognition,, image processing and e-learning..

.