Research Journal of Applied Sciences, Engineering and Technology 4(8): 877-883, 2012
ISSN: 2040-7467

Submitted: October 07, 2011 Accepted: November 16, 2011 Published: April 15, 2012

A New Loss-Tolerant Image Encryption Scheme Based on Secret Sharing and Two Chaotic Systems

[1]Li Li, [2]Ahmed A. Abd El-Latif, [2]Zhenfeng Shi and [1, 2]Xiamu Niu
[1]School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150080, China

Abstract: In this study, we propose an efficient loss-tolerant image encryption scheme that protects both confidentiality and loss-tolerance simultaneously in shadow images. In this scheme, we generate the key sequence based on two chaotic maps and then encrypt the image during the sharing phase based on Shamir's method. Experimental results show a better performance of the proposed scheme for different images than other methods from human vision. Security analysis confirms a high probability to resist both brute-force and collusion attacks.

Key words: Chaotic systems, loss-tolerant image encryption, secret sharing, security analysis

INTRODUCTION

Confidentiality and loss-tolerance are important properties for robust image transmission over unsecured channel. Robust image encryption methods based on Shamir's (*k*, *n*)-threshold method (Shamir, 1979) Eq. (1) has been studied (Thien and Lin, 2002; Lin and Lin, 2007; Shi *et al*., 2008; Yang *et al*., 2010; Zhao *et al*., 2009), which is known as secret image sharing. Any *k* of the *n* shadow images generated by those methods can reconstruct the original secret image. Thus it tolerates at most *n-k* shadow images faulty or lost, and the loss-tolerance property is guaranteed. However it could not provide confidentiality for the original image from human vision. The confidentiality is usually realized by the cryptographic techniques, such as permutation on the original image before sharing phase with the variable *X* in Shamir's polynomial defined by constant numbers (Thien and Lin, 2002; Lin and Lin, 2007; Yang *et al*., 2010) or generated by RSA algorithm (Zhao *et al*., 2009).

$$f(X) = a_0 + a_1 * X + \ldots + a_{k-1} * X^{k-1} \pmod{p} \quad (1)$$

In permutation method, it will permute the image pixels. Assume image pixels are permuted according to the increasing order of each value in the generated pseudorandom sequence which is generated based on chaotic system. It needs to operate the ordering algorithm firstly to sort each value in the pseudorandom sequence. Then the encryption method based on permutation and secret sharing such as (Thien and Lin, 2002; Lin and Lin, 2007; Yang and Ciou, 2010) consists of chaotic sequence generation, sorting, pixel permutation, and secret sharing. Thus it is not efficient even though *X* is defined by constant numbers in [1, …, *n*]. Considering the inefficiency for public cryptosystem (Paar,2005), the method based on RSA (Zhao *et al*., 2009) is also not efficient.

In this study, a new loss-tolerant image encryption scheme is proposed. In the new scheme, the shadow images are generated based on Shamir's (*k*, *n*) secret sharing with random numbers generated based on chaotic logistic map and *X* is generated based on chaotic tent map. It realizes the confidentiality and loss-tolerance simultaneously by adapting two chaotic maps in the sharing phase.

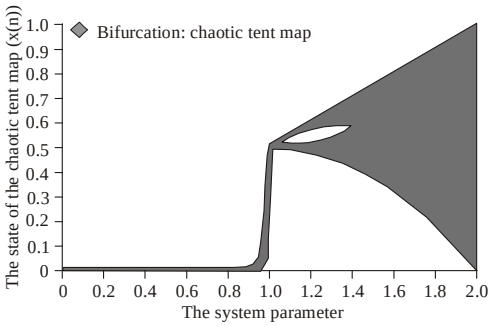
THE PROPOSED SCHEME

Chaotic systems:

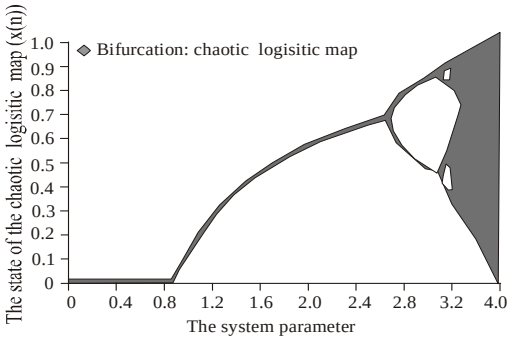
Chaotic tent map: Eqs. (2) and (3) define the chaotic tent map and x_m is the state value of the chaotic tent map (Amin *et al*., 2010). The researches have already shown that x_m has a uniform distribution (Amin *et al*., 2010). The bifurcation diagram describes the dynamics of chaotic tent map as shown in Fig. 1a. The continuous dash regions are the spaces in which the parameters can be used as valid keys:

$$T(x) = \begin{cases} rx & x \le 0.5 \\ r(1-x) & 0.5 < x < 1 \end{cases}, r \in [0,2] \quad (2)$$

Corresponding Author: Xiamu Niu, School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China, Tel.: +86- 451-86402861



(a)



(b)

Fig. 1: (a) Bifurcation behavior of the chaotic tent map, (b) Bifurcation behavior of the chaotic logistic map

$$x_m = T(x_{m-1}),\ x_m \in [0, 1] \quad (3)$$

Chaotic logistic map: Equation (4) defines the chaotic logistic map and y_m is the state value of the chaotic logistic map (Pareek *et al*., 2006). The researches have already shown that x_m has a uniform distribution (Pareek *et al*., 2006). Fig. 1b) shows the bifurcation behavior of chaotic logistic map:

$$y_m = r*y_{m-1}*(1-y_{m-1}),\ y_m \in [0, 1],\ r \in [0, 4] \quad (4)$$

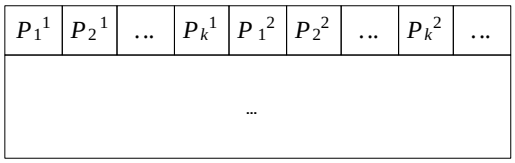


Fig. 3: Image partition

The proposed encryption scheme: The encryption diagram for the proposed method is shown in Fig. 2. The image with size $M*N$ (i.e. number of pixels) is first partitioned into $M*N/k$ non-overlapping blocks (block and section are used alternately in this paper) and each block has k pixels as shown in Fig. 3 where P_l^j denotes the lth pixel in block j. There are totally n shares for each block. All the ith shares in each block constitute the ith shadow image with size $M * N/k$ as shown in Fig. 4.

The polynomial applied in the image sharing phase is given in Eq. (5) evolved from Eq. (1) where R_i^j is the random number for computing the ith share $f_j(X_i^j)$ in block j. Two keys X_i^j and R_i^j are used to encrypt the pixel values. X_i^j is generated based on chaotic tent map (Amin *et al*., 2010) as in Eq. (2) and (3), while R_i^j is generated based on chaotic logistic map (Pareek *et al*., 2006) as in Eq. (4). Since x_m, $y_m \in [0, 1]$, and X_i^j, $R_i^j \in [0, 250]$, we obtain integral X_i^j and R_i^j by Eq. (6) and (7), respectively.

$$S_i^j = f_j(X_i^j) = P_1^j + P_2^j * X_i^j + \ldots P_k^j * X_i^{j(k-1)} + R_i^j \ (\text{mod } 251),\ P_l^j, X_i^j, R_i^j \in [0, 250],\ l \in [l, k]\ i \in [1, n],\ j \in [1, M*N/k] \quad (5)$$

$$X_i^j = Round\ (x_m * 250) \quad (6)$$

$$R_i^j = Round\ (y_m * 250) \quad (7)$$

The following steps are used to generate X_i^j and R_i^j for all the image blocks with the initial value $i = 1$, $j = 1$, $x_1 = x_0$, $y_1 = y_0$, $m_1 = m_0$. x_0, y_0, m_0, δ_1 and δ_2 will be initialized in the experiment.

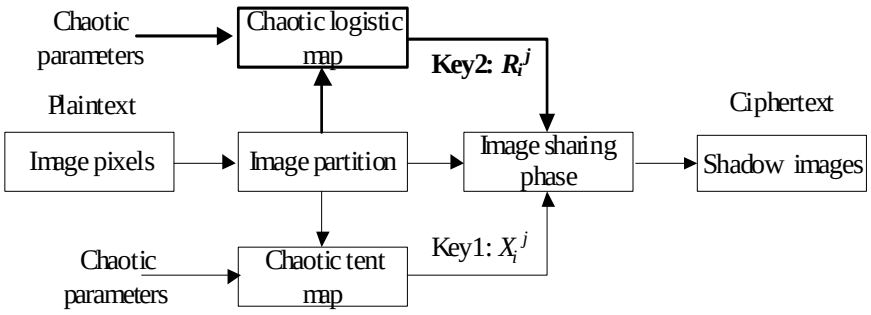


Fig. 2: Encryption diagram for the proposed method

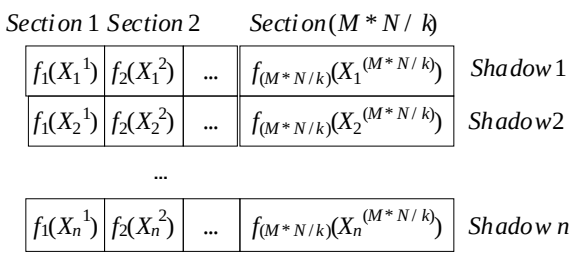


Fig. 4: Shadow images generation

From Eq. (4),we could know that y_m has the maximum number when $y_{m-1} = 0.5$.

Step 1: Repeat m_1 times with initial value x_1 to compute x_{m1} by Eq. (2) and (3). Repeat m_1 times with initial value y_1 to compute y_{m1} by Eq. (4). If j>M*N/k, go to step 5. If $m_1 = m_0 + n$, go to step 3

Step 2: Maps the real value x_{m1} into the domain [0, 250] to obtain X_i^j as Eq. (6); maps the real value y_{m1} into the domain [0, 250] to obtain R_i^j as Eq. (7); increase $m_1 = m_1 + 1$, $i = i + 1$, and then go to step 1

Step 3: The iteration times is set as $m_1 = m_0$. The initial value x_1 is increased by δ_1 as $x_1 = x_1 + \delta_1$; if $x_1 >= 1.0$, $x_1 = x_0$. The initial value y_1 is increased by δ_2 as $y_1 = y_1 + \delta_2$; if $y_1 > 0.5$, $y_1 = y_0$;

Step 4: Increase the block number as $j = j + 1$; go to step 1

Step 5: End

According to the above steps to generate different R_i^j i.e., R_1^j, R_2^j, …, R_n^j for all the n shares in each section j, and thus it will generate at most one of them equal to zero, e.g., $R_i^j = 0$. Consider the case in which neighboring image sections from section 1 to section s are black areas. Even though the pixel values $P_1^1 = P_2^1 = \ldots = P_k^1 = \ldots = P_1^s = P_2^s = \ldots = P_k^s = 0$, after adding different random number R_i^j in each polynomial for share i in section j, at most one value $f_j(X_i^j) = 0$ for all the n shares. Therefore, it will obtain different share value $S_1^j, \ldots, S_i^j, \ldots, S_n^j$ and only $S_i^j = 0$ in section j. Furthermore, it has a low probability for $S_i^1 = S_i^2 = \ldots = S_i^s = 0$ in the same shadow image as shown in Fig. 5. Thus it will not reveal any information about the secret image with large black neighboring areas.

From the steps in generating X_i^j and R_i^j, we can see that X_i^j and R_i^j with the same share number i in different blocks are generated with the same iteration number m_1 but different initial values i.e., x_1, $x_1 + \delta_1$, $\ldots x_1 + t_1 * \delta_1$, …, x_{max} and y_1, $y_1 + \delta_2$, $\ldots y_1 + t_2 * \delta_2$, …, y_{max}, respectively. This means that the distribution of pixel values in each shadow image is determined by value x_1, t_1, δ_1, x_{max} and y_1, t_2, δ_2, y_{max}. If $x_t = x_0 + t_1 * \delta > x_{max}$, $x_{t1} = x_0$. It is the same for y_{t2}. The value t_1 and t_2 in the initial value computation means that it will repeat the initial value x_0 and y_0 after t_1 and t_2 times respectively, and they influence the X_i^j and R_i^j reflected in the shadow image.

For the images with small area of black pixels, the value t_1 and t_2 influence slightly for the randomness of shadow image since $f_j(X_i^j)$ is determined by P_l^j, X_i^j and R_i^j. Even X_i^j and R_i^j in block j are the same as in block q, P_l^j and P_l^q are different and nonzero, thus $f_j(X_i^j)$ does not equal to $f_q(X_i^q)$. But it is not the same for images with large area of black pixels which have zero pixel values in several blocks. In the block with all zero pixel values, Eq. (5) degenerate to Eq. (8).

$$f_j(X_i^j) = R_i^j \pmod{251} \quad (8)$$

The corresponding pixel values $f_j(X_i^j)$ in the shadow image are only defined by R_i^j. In this case if t_2 is small, the pixel values $f_j(X_i^j)$ will repeat after a small image area which causes block effect and it is not random. Thus t_2 should be large enough to repeat previous pixel values only after a large image area in the same shadow image. For the inequality $y_0 + t_2 * \delta_2 > y_{max}$ with determined y_0 and y_{max}, if t_2 is large, δ_2 is small. Thus, the increasing number δ_2 should be small.

The decryption process is shown in Fig. 6. Each received shadow image i is first partitioned into $M*N / k$ sections. X_i^j and R_i^j are regenerated. The original image could only be correctly recovered with X_i^j, R_i^j and the shadow images during the image revealing phase. The revealing phase is the same as the Thien-Lin (2002) scheme with the obtained X_i^j. Some pixel values greater than 251 are recovered losslessly by adopting the method proposed in (Thien and Lin, 2002).

EXPERIMENTAL RESULTS AND ANALYSIS

Here, we do experiments for validating the security and practicability of the proposed algorithm. All the experiments are done by Visual C++6.0 in a computer of Intel Core i3 CPU@2.93 GHz and 3.36 GB of RAM.

To test the security and efficiency of the proposed scheme for different images such as homogenous,

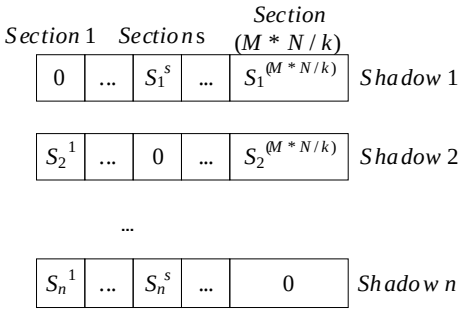


Fig. 5: An example for images with large area of black pixels using the proposed method

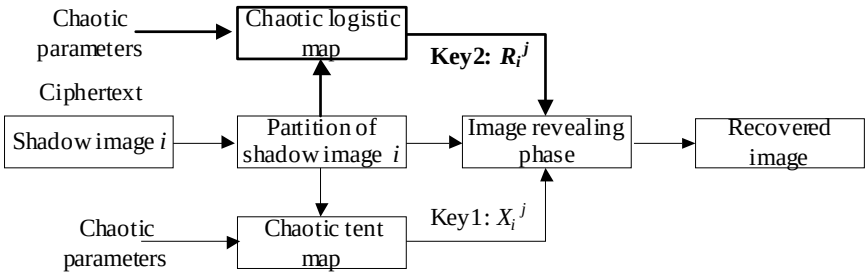


Fig. 6: Decryption diagram for the proposed method

Table 1: Parameters selected for logistic map

Images	r	y_0	m_0	δ_2
Fig. 8-9	3.999	0.01	100	0.01
Fig. 10-11	3.999	0.001	100	0.001
Fig. 12-13	3.999	0.001	100	0.0000001

repeated pattern, and low/high frequency, we use two standard grayscale images with size 256*256, such as *Lena* and *Rec* as shown in Fig. 7. Lena is image with low and high frequency, and Rec is homogeneous and has repeated patterns such as repeated black areas and white areas. And we adopt (2, 4) threshold. Thus, each shadow image has size 128*256. The proposed method applies chaotic tent map and chaotic logistic map to generate X_i^j and R_i^j, respectively. The initial parameters in chaotic tent map are selected as $r = 1.97$, $x_0 = 0.1$, $m_0 = 100$, and $\delta_1 = 0.01$. The initial parameters in chaotic logistic map with different precision are shown in Table 1.

Figure 8-9 show the shadow images for Lena and Rec by using the proposed method with logistic map parameter $r = 3.999$, $y_0 = 0.01$, $m_0 = 100$ and $\delta_2 = 0.01$. It has good randomness for Lena but reveals information about the black areas in original Rec image. Another two groups of experiments with logistic map parameter $r = 3.999$, $y_0 = 0.001$, $m_0 = 100$, $\delta_2 = 0.001$ and $r = 3.999$, $y_0 = 0.001$, $m_0 = 100$, $\delta_2 = 0.0000001$ obtain better performance as shown from Figure 10-13. Especially Fig. 13 shows the best randomness in the shadow images for the image *Rec* with large areas of black pixels. The experiments show that the proposed method could achieve a high confidentiality from human vision under a high chaotic map precision.

Since existing secret image sharing methods are mainly based on the method in (Thien and Lin, 2002), we implement two testing groups to compare the performance between the proposed scheme and (Thien and Lin, 2002) which realizes image sharing after permuting image pixels. The pseudorandom sequence for permutation is obtained by the same chaotic tent map used for generating X_i^j. And the pixels are permuted according to the increasing order of values in the generated sequence. Figs. 14-15 illustrate the shadow images obained by method (Thien and Lin, 2002). Figure 14-15 show that method (Thien and Lin, 2002) has approximate result from human vision but it seems blacker than the proposed method under the high precision $r = 3.999$, $y_0 = 0.001$, $m_0 = 100$, $\delta_2 = 0.0000001$ especially for Rec. And the information entropy for the shadow images generated by these two methods are computed and listed in Table 2. It is shown in Table 2 that the proposed method with high precision has better confidentiality. The recovered images for Lena and Rec are exactly the same as the original image in Figure 7 for all the methods.

Security analysis:

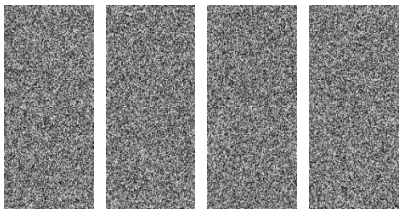
Brute-force attack: The attackers try to guess the key (i.e., all the X_i^j and R_i^j) in Eq. (1) using brute force. Each

Table 2: Information entropy comparisoin between the proposed method and Thien and Lin (2002)

Considered items	Lena using the proposed method	Lena using Thien and Lin (2002)	Rec using the proposed method	Rec using Thien and Lin (2002)
Shadow 1	7.9661	7.9629	7.9531	7.7713
Shadow 2	7.9653	7.9652	7.9522	7.7861
Shadow 3	7.9656	7.9655	7.9471	7.7851
Shadow 4	7.9664	7.9661	7.9428	7.8046

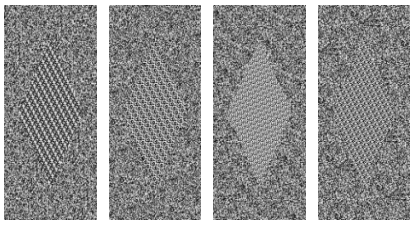
(a) (b)

Fig. 7: Original images (a) Lena, (b) Rec



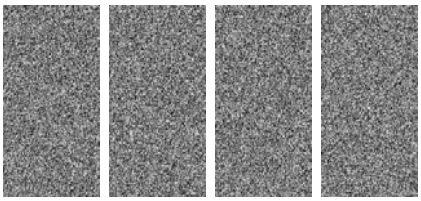
(a) (b) (c) (d)

Fig. 8: Shadow images for *Lena* with y_0 = 0.01 and δ_2= 0.01: (a)Shadow 1, (b)Shadow 2, (c)Shadow 3, (d)Shadow 4



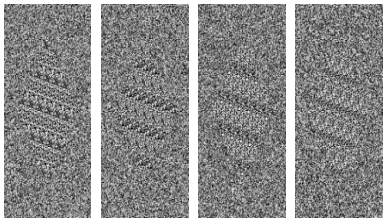
(a) (b) (c) (d)

Fig. 9: Shadow images for *Rec* with y_0 = 0.01 and δ_2 = 0.01: (a) Shadow 1, (b) Shadow 2, (c) Shadow 3, (d) Shadow 4



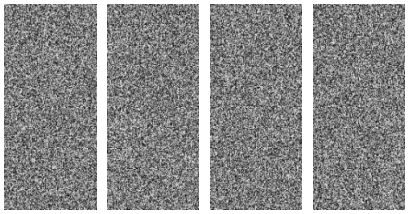
(a) (b) (c) (d)

Fig. 10: Shadow images for *Lena* with y_0 = 0.001 and δ_2 = 0. 001: (a) Shadow 1, (b) Shadow 2, (c) Shadow 3, (d) Shadow 4



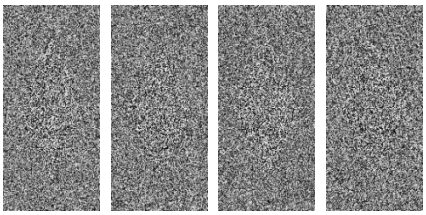
(a) (b) (c) (d)

Fig. 11: Shadow images for *Rec* with y_0 = 0.001 and δ_2 = 0. 001: (a) Shadow 1,(b) Shadow 2, (c) Shadow 3, (d) Shadow 4



(a) (b) (c) (d)

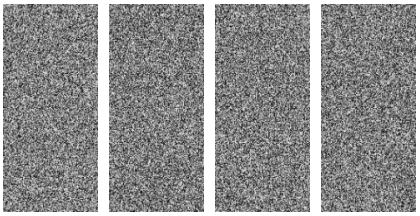
Fig. 12: Shadow images for *Lena* with y_0 = 0.001 and δ_2 = 0. 0000001: (a) Shadow 1, (b) Shadow 2, (c) Shadow 3, (d) Shadow 4



(a) (b) (c) (d)

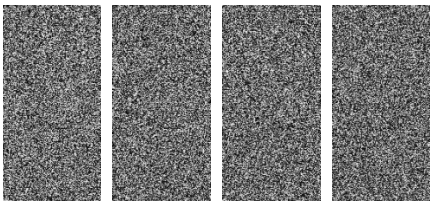
Fig. 13: Shadow images for Rec with y_0 = 0.001 and δ_2 = 0. 0000001: (a) Shadow 1, (b) Shadow 2, (c) Shadow 3, (d) Shadow 4

X_i^j, $R_i^j \in [0, 250]$ has totally 251possible values. It requires at least k shares to recover P_l^j, thus the attacker should guess at least k of $X_1^j,\ldots, X_n^j$ and k of $R_1^j,\ldots, R_n^j$ in block j. However, $X_1^j,\ldots, X_n^j$ are different from each other by the Interpolation theorem (Fang, 2008). Thus k of $X_1^j, \ldots, X_n^j$ has $P(251, k) = 251*250*\ldots*(251-k+1)$ possible values. Similarly, k of $R_1^j,\ldots, R_n^j$ has $P(251, k) = 251*250*\ldots*(251-k+1)$ possible values. There are totally $(P(251, k))^2$ possible values. Besides, X_i^j or R_i^j for



(a) (b) (c) (d)

Fig. 14: Shadow images for *Lena* generated by (Thien and Lin, 2002): (a) Shadow 1, (b) Shadow 2, (c) Shadow 3, (d) Shadow 4



(a) (b) (c) (d)

Fig. 15: Shadow images for *Rec* generated by (Thien and Lin, 2002): (a) Shadow 1, (b) Shadow 2, (c) Shadow 3, (d) Shadow 4

different blocks can be the same because they are in different polynomials. Therefore, the successful probability to recover the original image is only $1/P(251, k)^{(2*M*N/k)}$.

Collusion attack: Equation (5) demands at least k equations to solve the unique P_l^j (l = 1, 2,…, k) in each block; otherwise P_l^j has infinite solutions where corresponding X_i^j are known (Fang, 2008). In this paper, only the owner of the original image could generate X_i^j by Eq. (6), thus all the X_i^j are unknown to the conspirators. Assume K conspirators try to recover all the P_l^j. They have only shadow images {shadow1,…, shadow K} and reveal them to each other. Each conspirator brings in one unknown variable X_i^j as well as one known variable $f(X_i^j)$. We consider the best and worst case for the K conspirators where $K \in [2, n]$.

Best case: P_1^j,…, P_k^j are the same in each block. It needs to solve one P_l^j (l = 1, 2,…, k), K of R_1^j,…, R_n^j and K of X_1^j,…, X_n^j in each block; there are totally $2K + 1$ variables for K equations. Thus it has infinite solutions. They could only guess P_l^j with a successful probability 1/251 by brute force and thus a probability $(1/251)^{M*N/k}$ for all blocks.

Worst case: P_1^j,…, P_k^j are different between each other in each block. It needs to solve P_1^j,…, P_k^j, K of R_1^j,…, R_n^j and K of X_1^j,…, X_n^j in each block; there are totally $k + 2K$ variables for K equations. It also has infinite solutions. They guess all P_l^j in each block with a successful probability $(1/251)^k$ by brute force. The successful probability is $(1/251)^{k*(M*N/k)} = (1/251)^{M*N}$ for all blocks. The conspirators only succeed in a low probability between $(1/251)^{M*N}$ and $(1/251)^{M*N/k}$.

Therefore, the proposed method could resist the brute-force and collusion attacks in a high probability.

CONCLUSION

Efficient loss-tolerant image encryption scheme is proposed. The proposed method realizes both confidentiality and loss-tolerance in sharing images by adapting two keys based on two chaotic maps and then encrypt the image during the sharing phase based on Shamir's method. Experimental results show a better performance for different kinds of images, especially for homogenous and pattern repeated images. Security analysis shows that the proposed method is resistant against brute-force and collusion attacks.

ACKNOWLEDGMENT

This study is supported by the National Natural Science Foundation of China (Project Number: 60832010, 61100187).

REFERENCES

Amin, M., O.S. Faragallah and A.A. Abd El-Latif, 2010. A chaotic block cipher algorithm for image cryptosystems. Commun Nonlinear Sci Numer Simulat, 15: 3484-3497.

Fang, W.P., 2008. Secret image sharing safety. IEICE Proceeding on 14th Asia-Pacific IEEE International Conference Communications (APCC2008), Akihabara, Tokyo, Japan.

Lin, S.J. and J.C. Lin, 2007. VCPSS: A two-in-one two-decoding-options image sharing method Combining Visual Cryptography (VC) and Polynomial-Style Sharing (PSS) approaches. Pattern Recognition, 40: 3652-3666.

Paar, C., 2005. Applied cryptography and data security (Lecture notes). Ruhr-Universität Bochum. Retrieved from: http://www.crypto.ruhr-uni-bochum.de/ en_lectures.html.

Pareek, N.K., V. Patidar and K.K. Sud, 2006. Image encryption using chaotic logistic map. Image Vision Comp., 24: 926-934.

Shamir, A., 1979. How to share a secret. Communication ACM, 22(11): 612-613.

Shi, R.H., H. Zhong, L.S. Huang and Y.L. Luo, 2008. A (t, n) secret sharing scheme for image encryption. Congress on Image and Signal Processing [S. 1]: IEEE Computer Society.

Thien, C.C. and J.C. Lin, 2002. Secret image sharing. Comp. Graph., 26: 765-770.

Yang, C.N. and C.B. Ciou, 2010. Image secret sharing method with two-decoding-options: lossless recovery and previewing capability. Image Vision Comp., 28: 1600-1610.

Zhao, R., J.J. Zhao, F. Dai and F.Q. Zhao, 2009. A new image secret sharing scheme to identify cheaters. Comp. Standards Interfaces, 31(1): 252-257.