Cryptanalysis and Improvement of a Chaotic Map-Control-Based and the Plain Image-Related Cryptosystem

Bin Lu[1], Fenlin Liu[1], Xin Ge[1, *] and Zhenyu Li[2]

Abstract: Due to the characteristics of chaotic systems, different cryptosystems based on chaos have been proposed to satisfy the security of multimedia data. A plain image-related chaotic algorithm is proposed by Luo et al. with high speed and efficiency. Security weaknesses of the cryptosystem are studied in this paper. It is found that the important secret key information is leaked because an important parameter can be obtained after an inverse operation in the last step of the cryptosystems without secret key. Meanwhile, the value zero is processed improperly in quantification algorithm. Based on the weaknesses, chosen plaintext attack on the cryptosystem is proposed, by which, an important parameter, equivalent to secret key, can be calculated with a specific chosen plain image. With the obtained parameter, the plain image of any ciphered image, encrypted by the cryptosystem, can be recovered. Then, an improvement is proposed to solve the problems after modifying the quantification algorithm. It is from the experiments that chosen plaintext attack is valid and improved algorithm possesses better performance.

Keywords: Multimedia cryptosystem, cryptanalysis, chaos, chosen plaintext attack.

1 Introduction

With the continuous development of mobile communication technology, social networks are gradually integrating into people's life, which deeply affects our work habits, daily life and way of thinking. Images and videos, as important multimedia in social networks, involve a large number of important information about individuals and organizations. The protection of multimedia data has become an urgent problem for social network security. There are many ways to protect multimedia data, such as encryption [Fridrich (1998); Ge, Lu, Liu et al. (2016); Luo, Cao, Qiu et al. (2016)], information hiding [Xiong and Shi (2018)] and so on. Encryption as a basic data protection method has attracted more attention. Traditional cryptosystems possess relatively high security, but it is difficult to meet the requirements of massive data encryption in social networks due to their complex operations. The encryption technique based on chaos theory [Matthews (1989); Fridrich (1998)] is regarded to be a new choice considering its good cryptographic properties, such as the extreme sensitivity of the parameters, good pseudo randomness and so on.

[1] China National Digital Switching System Engineering Technology Research Center, Zhengzhou, China.

[2] Department of Computer Science, University of York, Deramor Lane, Heslington, UK.

[*] Corresponding Author: Xin Ge. Email: gexin_er@126.com.

CMC. doi:10.32604/cmc.2019.05633

Some cryptosystems based on chaos are proposed in order to improve the security of images [Hua and Zhou (2016); Chen, Mao and Chui (2004); Xiang, Hu and Sun (2015); Chai, Fu, Gan et al. (2019)] and videos [Lin, Yu, Lu et al. (2015)]. But some algorithms [Fridrich (1998); Lian (2009); Ye (2010)] are suffered from security problems due to important information leakage such as secret key information leakage, plain image information leakage. For examples, the hyper chaotic mage/video algorithm in [Lian (2009)] is broken due to the re-usage of keystream which equals to the secret key [Ge, Liu, Lu et al. (2011)]; Fridrich's chaotic image encryption [Fridrich (1998)] is broken by using influence network between cipher-pixels and the corresponding plain-pixels [Xie, Li, Yu et al. (2017)]; Ye's algorithm [Ye (2010)] is broken because the secret permutation does not change the values of the permuted elements [Li and Lo (2011)].

As one of the candidate algorithms, a chaotic map-control-based and the plain image-related cryptosystem [Luo, Cao, Qiu et al. (2016)] is proposed by Luo et al. (refers to as Luo algorithm in this paper), which is sensitive to both the secret key and the plain image, with high speed and efficiency. However, a key parameter generated from secret key and plaintext is embedded into the ciphertext, and it can be extracted after an inverse operation, which resulting in important information leaking of the secret key. Meanwhile, it is also a weakness by substituting zero with constant in quantification algorithm. In this paper, chosen plaintext attack on Luo algorithm is proposed with the secret key information leakage and the defect of the quantification algorithm. Then an improvement is proposed to solve the problem by improving the quantification algorithm and embedding the key parameter with secret key. Experiments and analysis show the validity of chosen plaintext attack and the security of improved algorithm.

The rest of the paper is organized as follows. Luo algorithm is briefly described in Section 2. A chosen plaintext attack is proposed after analyzing on the defect of Luo algorithm in Section 3. An improved algorithm is proposed in Section 4. Experimental results and analyses are reported in Section 5.

2 Luo algorithm

Luo algorithm is a single round encryption algorithm, including pixel value diffusion and pixel position scrambling. For an plain image $P=(p_{i,j})_{L\times H}$ of size $L\times H$, $p_{i,j}\in\{0,1,\cdots,255\}$, $i=1,2,\cdots,L$, $j=1,2,\cdots,H$, secret key $K=K_1,K_2,\cdots,K_N$ is N bytes, ciphered image of $P=(p_{i,j})_{L\times H}$ can be denoted by $C=(c_{i,j})_{L\times H}$. The procedure of Luo algorithm is briefly described below.

(1) Construct matrix $A=(a_{i,j})_{L\times(H+1)}$ with plain image $P=(p_{i,j})_{L\times H}$ and secret key K: pad the former H columns of A with plain image, and pad the $(H+1)$ th column of A with secret key bytes. Generate parameter I by applying quantification algorithm on A ($I=Q(A)$). Then generate parameter J by applying quantification algorithm on $K=K_1,K_2,\cdots,K_N$ ($J=Q(K)$).

Quantification algorithm $Q(M)$

For any matrix $M=(m_{i,j})_{W\times V}$, the quantification algorithm multiply each element of M and get a sequence of products noted as ME_j, $j=1,2,\cdots,W\times V$. Judge and ensure each value of ME_j be smaller than 10^{14}. ME_j is replaced by the multiplication of all digits in the decimal representation of ME_j when $ME_j>10^{14}$. To avoid the product equals to 0, digit number '0' is replaced by $\sqrt{2}$, $\sqrt{3}$, $\sqrt{5}$, $\sqrt{7}$, $\sqrt{11}$, $\sqrt{13}$, $\sqrt{17}$, $\sqrt{19}$, $\sqrt{29}$, $\sqrt{31}$, $\sqrt{37}$, $\sqrt{41}$ and $\sqrt{43}$ according to the position before multiplication. Let $D_1=ME_{W\times V}$. Then D_2 is calculated by multiplying the elements of M in reverse order with the above steps. Finally, calculate $Q(M)=\left((D_1+D_2)\bmod 10^{14}\right)\times 10^{-14}$.

(2) Calculate initial value x_0 of Tent map by $x_0=\begin{cases} I/J & \mathit{if}\ I<J \\ J/I & \mathit{otherwise}\end{cases}$. Iterate Tent map 200 times to avoid the non-randomness, continue to iterate Tent map and obtain a chaotic sequence $S=\{s_i\}_{i=1}^{L+H+8}$.

(3) Generate the pixel permutation vectors $\{row_i\}_{i=1}^{L}$, $\{column_i\}_{i=1}^{H}$ and the divisor parameter $k=\{k_0,\cdots k_7\}$ according $S=\{s_i\}_{i=1}^{L+H+8}$.

(4) Pixel value diffusion: Divide the $(8i+j)\ th$ pixel value p_{8i+j} of P by 2^{ki}, get the corresponding quotient v_{8i+j} and remainder r_{8i+j}. Pixel value q_{8i+j} of the image after the pixel value diffusion $Q=(q_{i,j})_{L\times H}$ is calculated by $q_{8i+j}=r_{8i+j}\times 2^{ki}+v_{8i+j}$.

(5) Pixel position permutation: Permute $Q=(q_{i,j})_{L\times H}$ with permutation vectors $\{row_i\}_{i=1}^{L}$ and $\{column_i\}_{i=1}^{H}$, denote the result by $D=(d_{i,j})_{L\times H}$, where $d_{i,j}=q_{row_i,column_j}$.

(6) Embed the parameter I into $D=(d_{i,j})_{L\times H}$: I can be denoted by $I=0.I_1\cdots I_8$ with eight integers $I_1,\cdots,I_8$ among 0 to 99 because I is a 10^{-15} precision decimal. Embed $I_1,\cdots,I_8$ into $D=(d_{i,j})_{L\times H}$ in fixed positions.

(7) Reprocess $D=(d_{i,j})_{L\times H}$ by a round of bit-XOR operation and obtain the ciphertext $C=(c_{i,j})_{L\times H}$, where $c_{1,1}=d_{1,1}$, $c_{i+1,1}=d_{i+1,1}\oplus c_{i,1}$, $c_{i,j+1}=d_{i,j+1}\oplus c_{i,j}$, $i=1,2,\cdots,H-1$, $j=1,2,\cdots,L-1$.

3 Cryptanalysis of Luo algorithm

In Luo algorithm, pixel value diffusion and pixel position permutation are both based on chaotic sequence. Considering chaotic system, its control parameter is public, initial value is calculated by I,J which are obtained by the quantification algorithm with secret key and plain image. If I,J are recovered by attacker, the initial value can be recovered, then the chaotic sequence can be obtained, and plain image can be correctly recovered. Therefore, key parameter I,J are important for the security of the algorithm.

3.1 Cryptanalysis on key parameter

Firstly, analysis on parameter I: According to Steps (6) and (7) of Luo algorithm, I is publicly embedded into image $D=(d_{i,j})_{L\times H}$ directly by substituting the corresponding values of fixed pixels, and $D=(d_{i,j})_{L\times H}$ can be obtained after a round of reverse bit-XOR operation on $C=(c_{i,j})_{L\times H}$. So the attacker can extract I without secret key.

Secondly, analysis on parameter J : parameter J and I are both calculated by quantification algorithm. The difference is that the calculation of J depends only on the secret key, while that of I depends not only on the secret key but also the plain image. If I has been obtained by the attacker, part information of J would also be leaked. And the leaking information of J is explored in **Proposition 1**.

Proposition 1. Parameter J satisfies $J=I$ if plain image $P=(p_{i,j})_{L\times H}$ satisfies $L=N$, $p_{i,j}=1$ for all $i=1,2,\cdots,L$, $j=1,2,\cdots,H$, where N is the number of secret key bytes $K=K_1K_2\cdots K_N$.

Proof: According to quantification algorithm in the generation of parameter J, $J=\left((E_1+E_2)\bmod 10^{14}\right)\times 10^{-14}$ where $E_1=K_1\otimes K_2\otimes\cdots\otimes K_N$, $E_2=K_N\otimes K_{N-1}\otimes\cdots\otimes K_1$, $\otimes$ denotes the multiplication defined in quantification algorithm.

According to the generation of parameter I, $I=Q(A)=\left((D_1+D_2)\bmod 10^{14}\right)\times 10^{-14}$, where $A=(a_{i,j})_{L\times(H+1)}$ (see Eq. (1)).

$$A=\left(\begin{array}{ccccc|c} p_{1,1} & p_{1,2} & \cdots & \cdots & p_{1,M} & K_1 \\ p_{2,1} & p_{2,2} & \cdots & \cdots & p_{2,M} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & K_N \\ p_{N+1,1} & p_{N+1,2} & \cdots & \cdots & \cdots & K_1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \end{array}\right) \tag{1}$$

If plain image $P=(p_{i,j})_{L\times H}$ satisfies $L=N$, $p_{i,j}=1$ for all $i=1,2,\cdots,L$, $j=1,2,\cdots,H$, where N is the number of secret key bytes $K=K_1K_2\cdots K_N$, then matrix $A=(a_{i,j})_{L\times(H+1)}$ equals to:

$$A=\left(\begin{array}{cccc|c} 1 & 1 & \cdots & 1 & K_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & K_{N-1} \\ 1 & 1 & \cdots & 1 & K_N \end{array}\right) \tag{2}$$

Then

$$D_1=1\otimes 1\otimes\cdots\otimes K_1\otimes 1\otimes\cdots\otimes K_2\otimes\cdots\otimes K_N=K_1\otimes K_2\otimes\cdots\otimes K_N$$

$$D_2=K_N\otimes 1\otimes\cdots\otimes K_{N-1}\otimes 1\otimes\cdots\otimes K_1\otimes\cdots\otimes 1=K_N\otimes K_{N-1}\cdots\otimes K_2\otimes K_1$$

So $E_1=D_1$ and $E_2=D_2$, therefore

$$J=\left((E_1+E_2)\bmod 10^{14}\right)\times 10^{-14}=\left((D_1+D_2)\bmod 10^{14}\right)\times 10^{-14}=I\ .\square$$

Note. According to **Proposition 1,** we can design a chosen plaintext attack, in which the chosen plain image satisfies the conditions of **Proposition 1**, the parameter J can be recovered from the ciphertext of the chosen plain image.

3.2 Chosen plaintext attack

According to the analysis above, a chosen plaintext attack (CPA) is proposed in this subsection. The goal of the attack is to recover the parameter J instead of secret key, and then any ciphered image can be correctly recovered with the parameter J. The details of CPA are shown as follows.

(1) Recover the key parameter J

(1.1) Construct a plain image $Q=(q_{i,j})_{N\times H}$, where N is the bytes number of secret key K, $q_{i,j}=1$ for all $i=1,2,\cdots,L$, $j=1,2,\cdots,H$. Input $Q=(q_{i,j})_{N\times H}$ into encryption machine, obtain the ciphered image $CT=(ct_{i,j})_{N\times H}$.

(1.2) Calculate $CT'=(ct'_{i,j})_{N\times H}$ with $CT=(ct_{i,j})_{N\times H}$ by $ct'_{i,j+1}=ct_{i,j+1}\oplus ct'_{i,j}$, $ct'_{i+1,1}=ct_{i+1,1}\oplus ct'_{i,1}$, $i=1,2,\cdots,H-1$, $j=1,2,\cdots,L-1$.

(1.3) Extract pixels' values from fixed position of $CT'=(ct'_{i,j})_{N\times H}$, and denote them by $a_1,\cdots a_8$, then recover the parameter $I'=0.a_1\cdots a_8$. So obtain $J=I'=0.a_1\cdots a_8$ according to **Proposition 1**.

(2) Recover any ciphered image with key parameter J:

(2.1) Calculate $C_{W\times V}$ with $C'_{W\times V}$ by $c'_{i,j+1}=c_{i,j+1}\oplus c'_{i,j}$ $c'_{i+1,1}=c_{i+1,1}\oplus c'_{i,1}$, $i=1,2,\cdots,W-1$, $j=1,2,\cdots,V-1$.

(2.2) Extract pixels' values from fixed position of $C'_{W\times V}$, denote them by $b_1,\cdots b_8$, and construct $I=0.b_1\cdots b_8$.

(2.3) Calculate the initial value x_0 of Tent map with I and J: $x_0=\begin{cases} I/J & J>I \\ J/I & otherwise \end{cases}$.

(2.4) Iterate Tent map with initial value x_0 for 200 times, obtain the chaotic sequence.

(2.5) Decrypt $C'_{W\times V}$ with the chaotic sequence, then the plain image $P_{W\times V}$ is recovered.

Note. According to **Proposition 1**, we know that $J=I'=0.a_1\cdots a_8$, so initial value x_0 obtained by the attack is correct. Therefore any ciphered image can be correctly recovered.

Although, to resist the CPA above, the cryptology designer could modify the encryption algorithm easily by refusing to encrypt the plain image whose row number equals to the number of secret key bytes. Security problem still exists because the quantification algorithm $Q(M)=Q(m_1,m_2,\cdots,m_n)$ deals with the value '0' improperly. According to the procedure of quantification algorithm, $D_1=m_1\otimes m_2\otimes\cdots\otimes m_n$. If m_i equals to 0, then $D_1=m_1\otimes m_2\otimes\cdots\otimes m_i\otimes m_{i+1}\otimes\cdots\otimes m_n$ $=m_1\otimes m_2\otimes\cdots\otimes 0\otimes m_{i+1}\otimes\cdots\otimes m_n$ $=c\otimes m_{i+1}\otimes\cdots\otimes m_n$, where c is a constant number. So another more complicated chosen plaintext attack algorithm can be designed.

4 Improvement of Luo algorithm

From the analyses in Section 3, Luo algorithm mainly has several weaknesses: (1) The parameter I can be obtained, and leaks part information of secret key. Especially, when plain image satisfies the conditions of **Proposition 1**, the key parameter J equals to I, which means that the information of secret key is leaked completely. (2) The inverse operation of the last step of Luo algorithm (step (7)) can be performed without secret key, so the step has no validity for security. (3) The quantification algorithm of Luo algorithm deals with the value ‘0’ improperly.

In order to solve the above problems, an improvement of Luo algorithm is proposed in the section. The improved algorithm is proposed after the modification of quantification algorithm.

4.1 Improvement of quantification algorithm

Let $M=(m_0,m_1,\cdots,m_{k-1})$ be one-dimensional vector, where $m_i \in \{0,1,2\cdots,255\}$. The procedure of the improved quantification algorithm ($\widetilde{Q}(M)$) is described below:

(1) $i \leftarrow 0$, $ME \leftarrow 1$, $p_0=2, p_1=3, p_2=5,\cdots$ are continuous prime numbers.

(2) If $m_i=0$, then $m_i \leftarrow \sqrt{p_j}$,where $j = i \bmod N_p$, N_p is the number of prime numbers in Step (1).

(3) $ME \leftarrow ME \times m_i$, if $ME \le 10^{14}$, go to Step (5).

(4) Let $d_{l-1}, d_{l-2},\cdots d_0$ is decimal representation of ME, and $ME \leftarrow \prod_{l=0}^{l-1} x_i$, where $x_i = \begin{cases} d_i & if\ d_i \neq 0 \\ \sqrt{p_i} & else \end{cases}$.

(5) $i \leftarrow i+1$, if $i=k$, go to Step (6), otherwise go to Step (2).

(6) $D_1 \leftarrow ME$, and the calculation of D_1 can be abbreviated as:

$$D_1 = \widetilde{Q}_{half}(M) = m_0 \otimes m_1 \otimes \cdots \otimes m_{L-1} \tag{3}$$

(7) The output of the quantification algorithm is $\widetilde{Q}(M)=\left((\widetilde{Q}_{half}(M)+\widetilde{Q}_{half}(M^T) \bmod 10^{14}\right)\times 10^{-14}$, where $M^T=(m_{k-1},m_{k-2},\cdots,m_0)$ is the inverse sequence of $M=(m_0,m_1,\cdots,m_{k-1})$.

In the improvement of quantification algorithm, irrational number $\sqrt{p_j}$ is introduced when $m_i=0$, in order to overcome the weakness that the quantification algorithm of Luo algorithm deals with the value ‘0’ improperly.

4.2 Improvement of Luo algorithm

In the improvement of Luo algorithm, the chaotic system, the control parameter and the secret key are the same as the original Luo algorithm. Let $Key=\{k_0,k_1,\cdots,k_{N-1}\}$ be the secret key, $P=(p_{i,j})_{L\times H}$ be the plain image of size $L\times H$, where pixel $p_{i,j}\in\{0,1,\cdots,255\}$, $i=1,2,\cdots,L$, $j=1,2,\cdots,H$. The improved Luo algorithm is described below:

(1) For secret key $Key=\{k_0,k_1,\cdots,k_{N-1}\}$, calculate $D_1=\widetilde{Q}_{half}(Key)=k_0\otimes k_1\otimes\cdots\otimes k_{N-1}$ and $D_2=\widetilde{Q}_{half}(Key^T)=k_{N-1}\otimes k_{N-2}\otimes\cdots\otimes k_0$ with the new quantification algorithm in Section 4.1. And then initial value $\widetilde{x}_0$ of Tent map can be calculated with Eq. (4):

$$\widetilde{x}_0=\begin{cases}D_1/D_2 & D_2>D_1\\ D_2/D_1 & otherwise\end{cases} \tag{4}$$

(2) Iterate Tent map with initial value $\widetilde{x}_0$ for n_b times(n_b is a large constant integer negotiated by encryption and decryption parties) to avoid the non-randomness, continue to iterate Tent map and obtain a chaotic sequence $\widetilde{X}=\{\widetilde{x}_i\}_{i=1}^{L+H+n_u}$,where $n_u=\left\lceil\dfrac{n_v\cdot\log_2(10)}{8}\right\rceil$, n_v is the maximum significant number of operations when implementation of encryption algorithm performed under finite precision operations, for example $n_v=15$.

(3) Calculate the parameter I and J with the new quantification algorithm in section 4.1. That is $I=\widetilde{Q}(P\,|\,Key)$ and $J=\widetilde{Q}(Key)$.

(4) Calculate another initial value x_0 of Tent map with Eq. (5).

$$x_0=\begin{cases}I/J & if\ I<J\\ J/I & otherwise\end{cases} \tag{5}$$

Iterate Tent map with initial value x_0 for n_c times (n_c is a large constant negotiated by encryption and decryption parties), and continue to iterate Tent map and obtain a chaotic sequence $S=\{s_i\}_{i=1}^{L+H+n_p}$ of $L+H+n_p$ length (n_p is the positive integer negotiated in advance, for example $n_p=8$).

(5) Sort $s_0,s_1,\cdots s_{L-1}$ in ascending order, and let the index of s_i in sorted sequence ($\{r_i\}_{i=0}^{L-1}$) be the row permutation vector. Similarly, sort $s_L,s_{L+1},\cdots s_{L+H-1}$ in ascending order, the column permutation vector $\{c_i\}_{i=0}^{H-1}$ can be obtained. And then calculate devisor parameter $dv=\{dv_0,\cdots dv_{n_p-1}\}$, where $dv_i=\lfloor 8\cdot s_{L+H+i}\rfloor$, $i=0,1,2,\cdots,n_p-1$.

(6) Convert $P=(p_{i,j})_{L\times H}$ into a one-dimensional vector $P=(p_0,p_1,\cdots,p_{L\times H-1})$, and then diffuse the pixel values with Eq. (6), denote the result of diffusion as $Q=(q_0,q_1,\cdots,q_{L\times H-1})$.

$$q_i=p_i<<dv_{i\bmod n_p} \tag{6}$$

where $i=0,1,\cdots,L\times H-1$, $<<$ is left cyclic shift operation. Convert $Q=(q_0,q_1,\cdots,q_{L\times H-1})$ into a two-dimensional matrix $Q=(q_{i,j})_{L\times H}$.

(7) Permute image $Q=(q_{i,j})_{L\times H}$ with $\{c_i\}_{i=0}^{H-1}$ and $\{r_i\}_{i=0}^{L-1}$, the permuted image $D=(d_{i,j})_{L\times H}$ is obtained, where $d_{i,j}=q_{r_i,c_j}$.

(8) Convert $D=(d_{i,j})_{L\times H}$ into a one-dimensional vector $D=(d_0,d_1,\cdots,d_{L\times H-1})$. For $I\in(0,1)$, convert $I\times 10^{n_v}$ into binary form (where n_v is the maximum significant number of operations), denote $I'_0,I'_1,\cdots,I'_u$ in bytes form from high to low, where $u=\left\lceil\frac{n_v\cdot\log_2(10)}{8}\right\rceil$. Calculate $pos_i=\lfloor\tilde{x}_i\cdot(L\times H-1)\rfloor$, the embedding position of parameter I. Replace the pixel d_{pos_i} by I'_i for $i=0,1,\cdots,u$. Convert $D=(d_0,d_1,\cdots,d_{L\times H-1})$ into a two-dimensional matrix $D=(d_{i,j})_{L\times H}$ again.

(9) Calculate $T=(t_0,\cdots t_{L+H\text{-}1})$ with chaotic sequence $\tilde{x}_u,\cdots\tilde{x}_{u+L+\text{H-1}}$ produced in step (1), where $t_i=\lfloor\tilde{x}_{u+i}\cdot 256\rfloor$, and the ciphertext $C=(c_{i,j})_{L\times H}$ can be obtained by Eq. (7).

$$c_{i,j}=d_{i,j}\oplus t_i\oplus t_{L+j} \tag{7}$$

Decryption procedure is the inverse procedure of encryption.

In the improved algorithm, another chaotic sequence $\tilde{X}$, generated with initial value $\tilde{x}_0$ ($\tilde{x}_0$ is calculated from secret key), is introduced. The first part of $\tilde{X}$ is used to produce the embedding positions of key parameter I, as a result, the positions are determined by secret key, and the attacker cannot get I without secret key, which guards against the information leakages of secret key. The second part of $\tilde{X}$ is used to diffuse the pixel values (in Step (9)), thus the attacker can't reverse Step (9) without secret key, which is useful to resist statistical analysis attacks and differential attacks. In summary, the improved algorithm is more secure than the original Luo algorithm.

5 Experiments

In this section, the correctness of chosen plaintext attack on Luo algorithm is tested, and experiments on the improved algorithm is carried out. Experiments are performed with double precision floating point numbers represented by 64 bits specified in IEEE on the PC in the Matlab2013a environment.

5.1 Experiments on CPA

According to section 3, CPA is proposed on Luo algorithm. Correctness of the attack is verified by experiments below.

Luo algorithm is first implemented, where secret key Key is 64 bytes generated randomly, control parameter of Tent map is 0.76589, and the embedding positions of parameter I are $p_{20,20},p_{20,21},\cdots,p_{20,27}$. Fig. 1(a) of size 512×512 is encrypted with Luo algorithm, and the ciphered image is shown in Fig. 1(b).

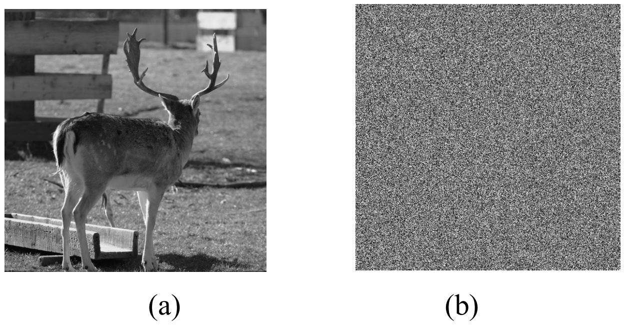


Figure 1: Plain image and its ciphered image

CPA on Luo algorithm is then implemented with chosen plain image of 64×64 (shown in Fig. 2(a)). According to the procedure of CPA, Fig. 2(a) is encrypted with Luo algorithm and the ciphered image is shown in Fig. 2(b), then key parameter J is obtained 0.025167271329850.

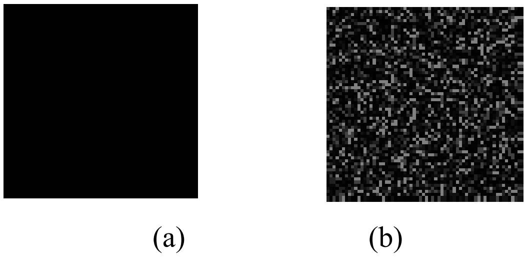


Figure 2: Chosen image and its ciphered image

With key parameter J , the plain image of any ciphered image encrypted with Luo algorithm can be correctly recovered. The ciphered image in Fig. 1(b) is taken as an example, and the recovered plain image is shown in Fig. 3. It is clear that the chosen plaintext attack in Section 3 is correct.

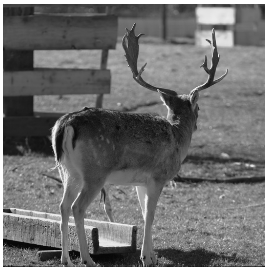
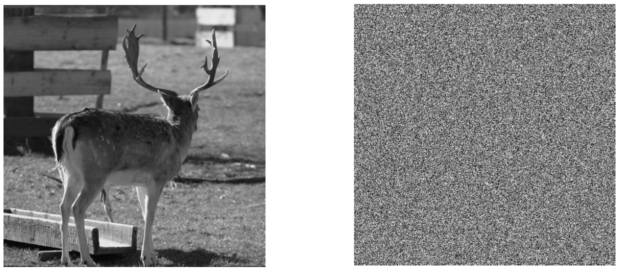


Figure 3: Result of chosen plaintext attack

5.2 Experiments on improved algorithm

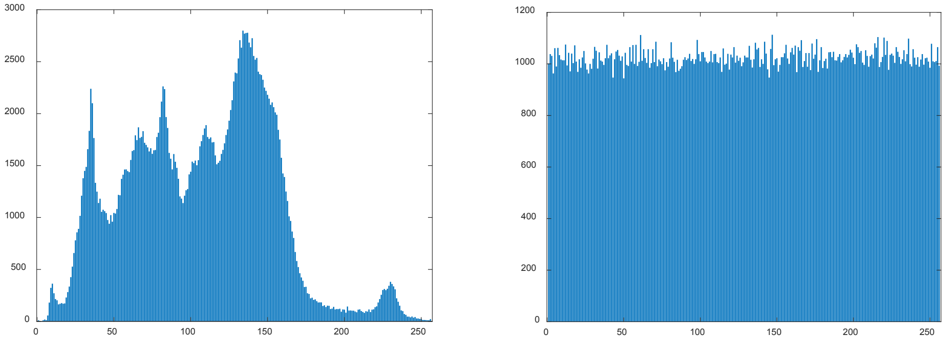
Experimental results on improvement are carried out in this section. In the experiments, secret key Key is 64 bytes generated randomly, control parameter of Tent map is 0.76589.

The plain and ciphered image is shown in Fig. 4(a) and Fig. 4(b) respectively. The histogram of plain image is shown in Fig. 5(a), and that of ciphered image is shown in Fig. 5(b). It is obvious that the histogram of ciphered image has a more uniform distribution.



(a) (b)

Figure 4: Plain and ciphered image with improved algorithm



(a) (b)

Figure 5: Histogram of plain image and ciphered image with improved algorithm

(1) Statistical analysis

Generally, high correlation exists among adjacent pixels for natural images in horizontal, vertical and diagonal directions. Therefore, the correlation between two adjacent pixels should be reduced in the ciphered image to withstand statistical attack. The correlation can be measured by correlation coefficient, which can be calculated by Eq. (8).

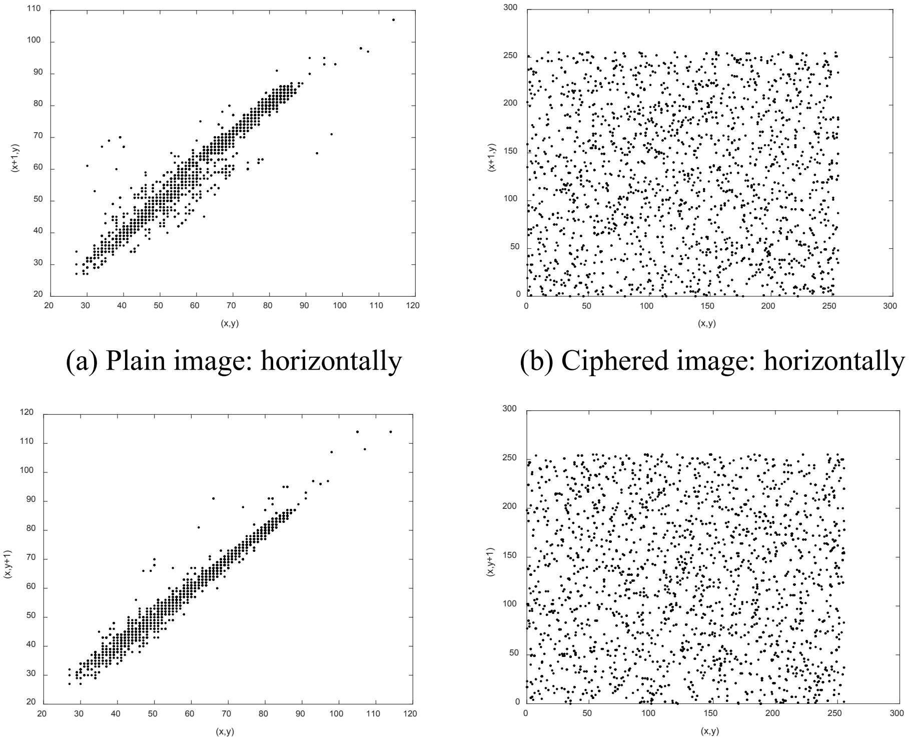
$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)D(y)}} \tag{8}$$

where x and y are gray-scale values of two adjacent pixels, and, $D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x)^2)$, $\mathrm{cov}(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)$, $E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i$.

It is expected that the correlation coefficient of plain image should be close to 1 while that of the ciphered image should be close to 0.

In experiments, 4096 pairs of two adjacent pixels of image in three directions are randomly selected for the calculation of the correlation coefficients.

Taking image “Deer” (Fig. 4(a)) and its ciphered image (Fig. 4(b)) as an example, the distributions of randomly selected 4096 pairs of two adjacent pixels in three directions are shown in Fig. 6 respectively. And then correlation coefficients of plain and ciphered image are calculated. The correlation coefficients of plain image for horizontally, vertically and diagonally adjacent pixels are 0.948904, 0.979602 and 0.929812 respectively; and those of ciphered image are -0.003246, -0.034536 and -0.011568 respectively. According to the results, we find that the correlation coefficients between the pixels in three directions after encryption by with the improved algorithm are greatly reduced, and the correlation is destroyed to a great extent.

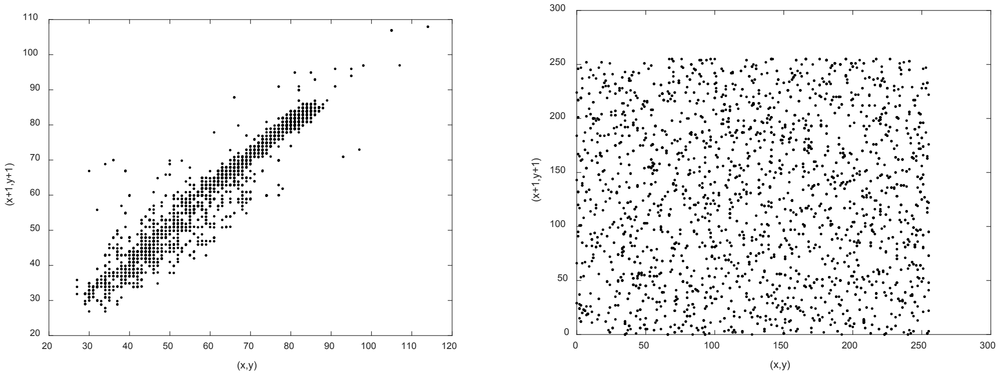


(a) Plain image: horizontally

(b) Ciphered image: horizontally

(c) Plain image: vertically

(d) Ciphered image: vertically

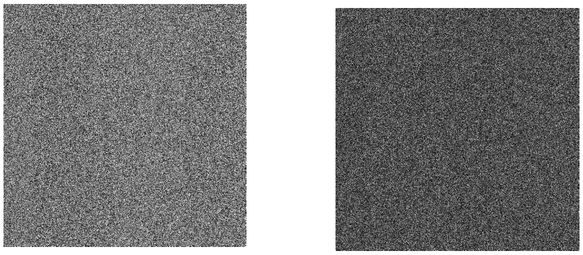


(e) Plain image: diagonally (f) Ciphered image: diagonally

Figure 6: Correlation distributions of adjacent pixels in plain and ciphered images

(2) Key sensitivity test

For key sensitivity test, we make a tiny change in the secret key, change one bit of the secret key and keep other parameters unchanged, then decrypt the ciphered image with the changed key. The experimental results are shown in Fig. 7. Decryption result with the changed key is shown in Fig. 7(a) and difference between decryption with the changed and original key is shown in Fig. 7(b). The rate of different pixels in the two images is about 93%, which means that nothing can be recovered as long as the attacker has tiny error of secret key. Therefore the improved algorithm is sensitive to the change of secret key.



(a) (b)

Figure 7: Key sensitivity test

6 Conclusions

In this paper, we study the security weaknesses of Luo algorithm, important information leaking of the secret key is found because a key parameter I can be extracted after an inverse operation. And it is also a weakness by substituting zero with constant in quantification algorithm. Based on the weaknesses, CPA attack on the cryptosystem is proposed. Meanwhile, to solve the problems, an improvement is proposed after modifying the quantification algorithm. Finally, experiments and analyses show the validity of chosen plaintext attack and the security of improved algorithm.

Acknowledgement: The work described in this paper was partially supported by the National Natural Science Foundation of China (Grant No. 61601517), basic and advanced technology research project of Henan Province, China (Grant No. 2014302703).

References

Chai, X. L.; Fu, X. L.; Gan, Z. H.; Lu, Y.; Chen, Y. R. (2019): A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Processing*, vol. 155, pp. 44-62.

Chen, G. R.; Mao, Y. B.; Chui, C. K. (2004): A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons & Fractals*, vol. 21, no. 3, pp. 749-761.

Fridrich, J. (1998): Symmetric ciphers based on two-dimensional chaotic maps. *International Journal of Bifurcation and Chaos*, vol. 8, no. 6, pp. 1259-1284.

Ge, X.; Liu, F. L.; Lu, B.; Wang, W. (2011): Cryptanalysis on a spatiotemporal chaotic image/video cryptosystem and its improved version. *Physics Letters A*, vol. 375, pp. 809-913.

Ge, X.; Lu, B.; Liu, F. L.; Gong, D. F. (2016): An image encryption algorithm based on information hiding. *International Journal of Bifurcation and Chaos*, vol. 26, no. 11, 1650192.

Hua, Z.; Zhou, Y. (2016): Image encryption using 2D Logistic-Adjusted-Sine Map. *Information Sciences*, vol. 339, pp. 237-253.

Lian, S. G. (2009): Efficient image or video encryption based on spatiotemporal chaos system. *Chaos, Solitons & Fractals*, vol. 40, no. 5, pp. 2509-2519.

Li, C.; Lo, K. T. (2011): Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Processing*, vol. 91, no. 4, pp. 949-954.

Lin, Z.; Yu, S.; Lu, J.; Cai, S.; Chen, G. (2015): Design and ARM-Embedded implementation of a chaotic map-based real-time secure video communication system. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 7, pp. 1203-1216.

Luo, Y. L.; Cao, L. C.; Qiu, S. H.; Lin, H; Harkin, J. et al. (2016): A chaotic map-control-based and the plain image-related cryptosystem. *Nonlinear Dynamic*, vol. 83, pp. 2293-2310.

Matthews, R. (1989): On the derivation of a "chaotic" encryption algorithm. C*ryptologia*, vol. 13, no. 1, pp. 29-42.

Xiang, T.; Hu, J.; Sun, J. (2015): Outsourcing chaotic selective image encryption to the cloud with steganography. *Digital Signal Processing*, vol. 43, pp. 28-37.

Xie, E. Y.; Li, C. Q.; Yu, S. M.; Lv, J. H. (2017): On the cryptanalysis of Fridrich's chaotic image encryption scheme. *Signal Processing*, vol. 132, pp. 150-154.

Xiong, L. Z.; Shi, Y. Q. (2018): On the privacy-preserving outsourcing scheme of reversible data hiding over encrypted image data in cloud computing. *Computers, Materials & Continua*, vol. 55, no. 3, pp. 523-539.

Ye, G. D. (2010): Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recognition Letters*, vol. 31, no. 5, pp. 347-354.