EURASIP Journal on Applied Signal Processing 2005:8, 1277–1288

Cryptanalysis of a New Signal Security System for Multimedia Data Transmission

Chengqing Li
Department of Mathematics, Zhejiang University, Hangzhou 310027, China
Email: swiftsheep@hotmail.com

Shujun Li
Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong
Email: hooklee@mail.com

Guanrong Chen
Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong
Email: eegchen@cityu.edu.hk

Gang Chen
Department of Mathematics, Zhejiang University, Hangzhou 310027, China
Email: gangchen@zju.edu.cn

Lei Hu
State Key Laboratory of Information Security, Graduate School of Chinese Academy of Sciences, Beijing 100049, China
Email: hu@is.ac.cn

Received 23 August 2004; Revised 4 January 2005; Recommended for Publication by C. C. Jay Kuo

Recently, a new signal security system called TDCEA (two-dimensional circulation encryption algorithm) was proposed for real-time multimedia data transmission. This paper gives a comprehensive analysis on the security of TDCEA. The following security problems are found: (1) there exist some essential security defects in TDCEA; (2) two known-plaintext attacks can break TDCEA; (3) the chosen-plaintext and chosen-ciphertext versions of the aforementioned two known-plaintext attacks can break TDCEA even with a smaller complexity and a better performance. Some experiments are given to show the security defects of TDCEA and the feasibility of the proposed known-plaintext attacks. As a conclusion, TDCEA is not suitable for applications that require a high level of security.

Keywords and phrases: TDCEA, image encryption, chaos, cryptanalysis, known/chosen-plaintext attack, multimedia.

1. INTRODUCTION

In today's digital world, the security of multimedia data, for example, digital speech, image and video files, becomes more and more important due to their frequent transmission over open networks. In some real applications, such as pay-TV, medical imaging systems, military image/database communications, and confidential video conferences, highly secure and reliable storage and transmission of multimedia data are needed. To fulfill such a demand, many encryption schemes have been proposed as possible solutions [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. Meanwhile, cryptanalysis work has also been developed, and some of the proposed schemes have been found to be insecure [9, 10, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]. For a comprehensive survey of the state-of-the-art of image and video encryption, see [26].

The present paper focuses on a new signal security system recently proposed in [1, 2], which is called the two-dimensional circulation encryption algorithm (TDCEA). In fact, TDCEA is an enhanced version of a previous image encryption scheme proposed by the same authors in [3, 4], named BRIE (bit recirculation image encryption), which is the 1D counterpart of TDCEA. The original BRIE scheme has been successfully cryptanalyzed in [19], showing its insecurity against known-/chosen-plaintext attacks. Although

TDCEA is more complicated than BRIE by using 2D permutations, this paper will point out that such a 2D generalization cannot enhance the security of BRIE against known-/chosen-plaintext and chosen-ciphertext attacks. In addition, it will be shown that the security of TDCEA against brute-force attack was much overestimated in [1, 2]. Essentially, TDCEA is a permutation-only image cipher, which has been known to be insecure against known-/chosen-plaintext attacks [15, 16, 23, 25].

The rest of this paper is organized as follows. The next section briefly introduces TDCEA and its 1D version BRIE. Section 3 discusses some general security defects of TDCEA. Two known-plaintext and chosen-plaintext attacks are given in Sections 4 and 5, respectively, with some experimental results for verification. Section 6 briefly discusses the chosen-ciphertext attack, a natural and simple generalization of the chosen-plaintext attack. Section 7 concludes the paper.

2. TDCEA

The basic idea used in TDCEA is secret bit rotations of every 64 consecutive bits (of 8 consecutive pixels), which are controlled by a chaotic pseudorandom binary sequence (PRBS). BRIE is the simplified version of TDCEA, by rotating only 8 bits in each pixel. To facilitate the description of TDCEA and BRIE, it is assumed that the plain image has size $M \times N$, where M is the height and N is the width of the image.

2.1. Definitions and notations

First, some definitions and notations are given in order to introduce TDCEA and BRIE. Assuming two matrices M and M' of size $m \times n$, where m is the height and n is the width, two mapping operations are defined as follows.

(i) *The horizontal rotation mapping*, $\text{Rotate}\, X_i^{p,r} : M \to M'$ $(0 \le i \le m-1)$, is defined to circularly rotate the ith row of M, in the left (when $p = 1$) or right (when $p = 0$) direction, by r elements.

(ii) *The vertical rotation mapping*, $\text{Rotate}\, Y_j^{q,s} : M \to M'$ $(0 \le j \le n-1)$, is defined to circularly rotate the jth column of M, in the up (when $q = 1$) or down (when $q = 0$) direction, by s elements.

When M is a $1 \times n$ vector, the 1D version of the above 2D rotation mapping is denoted by $\text{ROLR}_p^q : M \to M'$, which is defined to circularly rotate M in the left (when $p = 0$) or right (when $p = 1$) direction, by q elements.

2.2. The 1D version of TDCEA-BRIE

Assuming the plain image is $f = [f(x,y)]_{x=0,y=0}^{M-1,N-1}$ and the cipher image is $f' = [f'(x,y)]_{x=0,y=0}^{M-1,N-1}$, BRIE is described as follows [3, 4].

(i) *The secret key*: two integers α, β, and the initial condition $x(0) \in (0,1)$ of the following chaotic logistic map:

$$x(k+1) = \mu \cdot x(k) \cdot (1 - x(k)). \tag{1}$$

(ii) *The initialization procedure*: run the chaotic logistic map from $x(0)$ to generate a chaotic sequence, $\{x(k)\}_{k=0}^{\lceil (MN+1)/8 \rceil - 1}$, where $\lceil a \rceil$ denotes the smallest integer that is not less than a. From the 8-bit binary representation of $x(k)$ as

$$\begin{aligned} x(k) &= \sum_{i=0}^{7} b(8k+i) \cdot 2^{-i-1} \\ &= 0.b(8k+0)b(8k+1)\cdots b(8k+7), \end{aligned} \tag{2}$$

a PRBS is derived, $\{b(k)\}_{k=0}^{MN}$.

(iii) *The encryption procedure*: for the plain pixel $f(x,y) = \sum_{i=0}^{7} b_i \cdot 2^i$, the corresponding cipher pixel $f'(x,y) = \sum_{i=0}^{7} b'_i \cdot 2^i$ is determined by the following equation:

$$M' = \text{ROLR}_p^q(M), \tag{3}$$

where $p = b(N \cdot x + y)$, $q = \alpha + \beta \cdot b(N \cdot x + y + 1)$, and M, M' are two 1×8 bit matrices: $M = [b_7, b_6, \ldots, b_0]$, $M' = [b'_7, b'_6, \ldots, b'_0]$.

(iv) *The decryption procedure* is denoted by

$$M = \text{ROLR}_{1-p}^q(M') = \text{ROLR}_p^{8-q}(M'). \tag{4}$$

In [19], BRIE was successfully cryptanalyzed and the following security problems were pointed out.

(1) The key space is too small and the security against the brute-force attack was much overestimated.

(2) There exist some essential defects, which makes it possible for an attacker to get some visual information of the plain image by observing the cipher image.

(3) BRIE is not secure against known-/chosen-plaintext attacks, since only one known/chosen plain image is enough to get an equivalent key, a mask array $Q = [q(x,y)]_{x=0,y=0}^{M-1,N-1}$, where $q(x,y)$ satisfies

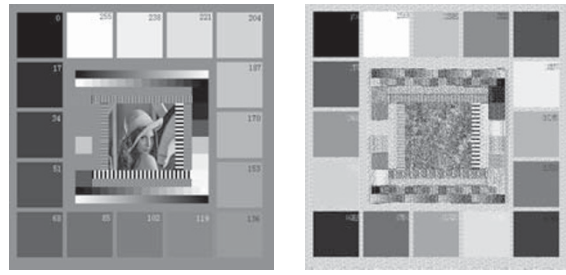
$$\begin{aligned} M' &= \text{ROLR}_0^{q(x,y)}(M), \\ M &= \text{ROLR}_1^{q(x,y)}(M') = \text{ROLR}_0^{8-q(x,y)}(M'). \end{aligned} \tag{5}$$

(4) It is easy to get the subkeys α, β and the most significant 8 bits of the chaotic state $x(k)$, as a replacement of the subkey $x(0)$, from the mask array Q obtained above.

2.3. TDCEA

TDCEA [1, 2] is an enhanced version of BRIE, by extending the bit rotation operations from one pixel to 8 consecutive pixels, and from two directions (left and right) to four directions (left, right, up, and down).

TDCEA encrypts a plain image block by block, where each block contains 8 consecutive pixels. To simplify the following description, without loss of generality, assume that MN can be divided by 8. Consider the 2D plain image $\{f(x,y)\}_{x=0,y=0}^{M-1,N-1}$ as a 1D signal $\{f(l)\}_{l=0}^{MN-1}$ by scanning it



(a) (b)

Figure 1: A special test image, "test pattern," encrypted by TDCEA. (a) The plain image. (b) The cipher image.

in raster order.[1] Then, the plain image can be divided into $MN/8$ blocks:

$$\left\{f^{(8)}(0),\ldots,f^{(8)}(k),\ldots,f^{(8)}\left(\frac{MN}{8}-1\right)\right\}, \tag{6}$$

where

$$f^{(8)}(k)=\{f(8k+0),\ldots,f(8k+i),\ldots,f(8k+7)\}. \tag{7}$$

Rewrite each block $f^{(8)}(k)$ as an 8×8 bit matrix $M_k=[M_k(i,j)]_{i=0,j=0}^{7,7}$, by assigning the 64 bits in the current block in the raster order $f(8k+i)=\sum_{j=0}^{7}M_k(i,j)\cdot 2^j$. In the same way, the 8 pixels of each block of the cipher image can be represented by an 8×8 bit matrix $M'_k=[M'_k(i,j)]_{i=0,j=0}^{7,7}$, where $f'(8k+i)=\sum_{j=0}^{7}M'_k(i,j)\cdot 2^j$. Based on the matrix representations of the plain/cipher images, the working mechanism of TDCEA can be described as follows.

(i) *The secret key*: two integers α, β, the initial condition $x(0)$, and the control parameter μ of the logistic map (1), where $0<\alpha<8$, $0\le\beta<8$ and $0<\alpha+\beta<8$.

(ii) *The initialization procedure*: run the logistic map starting from $x(0)$ to generate a chaotic sequence, $\{x(k)\}_{k=0}^{MN/8-1}$, and then extract the 17-bit representation of $x(k)$ to yield a PRBS, $\{b(i)\}_{i=0}^{17MN/8-1}$. In the hardware implementation given in [1, 2], the logistic map is realized in 17-bit fixed-point arithmetic.

(iii) *The encryption procedure.*

Step (1). Horizontal rotations: for $i=0\sim 7$ (i.e., for each value of i from 0 to 7, the same hereinafter) do $M_k^*=\text{Rotate}\,X_i^{p,r}(M_k)$, where $p=b(17k+i)$ and $r=\alpha+\beta\cdot b(17k+i+1)$.

Step (2). Vertical rotations: for $j=0\sim 7$ do $M'_k=\text{Rotate}\,Y_j^{q,s}(M_k^*)$, where $q=b(17k+8+j)$ and $s=\alpha+\beta\cdot b(17k+9+j)$.

(iv) *The decryption procedure* is a simple reversion of the above encryption procedure, as follows.

Step (1). Vertical rotations: for $j=0\sim 7$ do $M_k^*=\text{Rotate}\,Y_j^{q,s}(M'_k)$, where $q=1-b(17k+8+j)$ and $s=\alpha+\beta\cdot b(17k+9+j)$.

Step (2). Horizontal rotations: for $i=0\sim 7$ do $M_k=\text{Rotate}\,X_i^{p,r}(M_k^*)$, where $p=1-b(17k+i)$ and $r=\alpha+\beta\cdot b(17k+i+1)$.

3. SOME SECURITY DEFECTS OF TDCEA

3.1. Essential defects of circulations

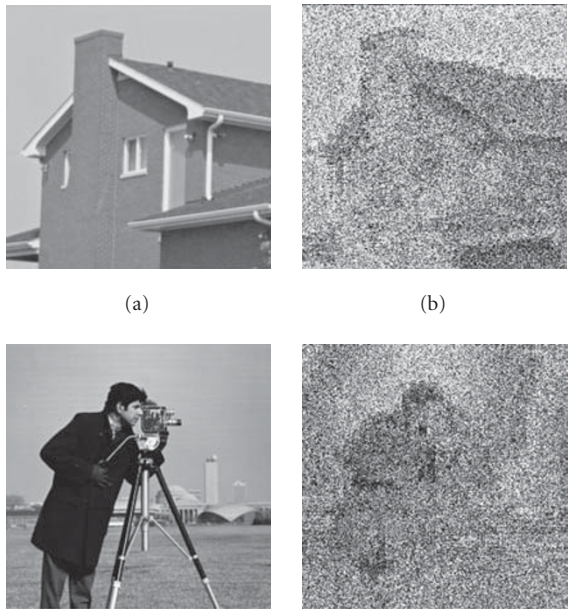
In [19], some essential defects of the ROLR operation were found: (1) some plain pixels may keep unchanged after encryption, so the plain image will roughly emerge if there are too many such pixels; (2) for a subregion in the plain image with a fixed gray value, at most eight gray values[2] will be contained in the corresponding subregion of the cipher image, which will lead the edge of this subregion to appear in the cipher image. The second fact is also true for subregions with close pixel values.

Although TDCEA extends the shift operation to two dimensions, the above defects of ROLR cannot be completely removed. As an extreme example, when all elements in M_k are 0-bits or 1-bits, it is obvious that $M'_k\equiv M_k$, which means TDCEA cannot encrypt blocks with fixed pixel value 0 (black) or 255 (white) at all. To test the performance of TDCEA compared with BRIE, we have encrypted the same test image used in [19] for BRIE, with the following parameters: $(\alpha,\beta)=(2,4)$, $x(0)=34816/2^{17}\approx 0.2656$, $\mu=128317/2^{15}\approx 3.9159$. The encryption result is shown in Figure 1, from which one can see that the 16 squares in the plain image remain fixed in the cipher image, though the fixed gray values have been changed for most squares. Comparing this result with those given in [19, Figure 1], it is obvious that the security defects of BRIE are not enhanced by TDCEA.

As a second example to test the possible enhancement of TDCEA on the BRIE security, we also tested the encryption performance of TDCEA on some general natural images

[1] Note that in [1, 2] TDCEA is described directly for 1D signals. In this paper, we prefer to explicitly mention the transform from 2D images to 1D signals, so as to emphasize the relation between BRIE and TDCEA (which is not mentioned in [1, 2]).

[2] For some pixel values, the number of different cipher pixel values is even smaller, which may be 1, 2, or 4 [19, Section 3.1].



(a) (b)

(c) (d)

Figure 2: Two natural images, House and Cameraman, encrypted by TDCEA, with $(\alpha,\beta) = (5,1)$, $x(0) = 33578/2^{17} \approx 0.2562$, and $\mu = 129518/2^{15} \approx 3.9526$. (a) House. (b) Encrypted House. (c) Cameraman. (d) Encrypted Cameraman.

containing many smooth areas. As known, the pixels within a smooth area generally have close pixel values, which are found similar to the squares with fixed gray values shown in Figure 1 when TDCEA is applied for encryption. Two images, "House" and "Cameraman," are selected for testing. The experimental results are shown in Figure 2, from which one can see many important edges of the plain images emerging in the cipher images. In this experiment, the parameters of TDCEA are as follows: $(\alpha,\beta) = (5,1)$, $x(0) = 33578/2^{17} \approx 0.2562$, and $\mu = 129518/2^{15} \approx 3.9526$.

3.2. Security problem of α, β

In [1, 2], the values of α and β are constrained by $0 < \alpha < 8$, $0 \le \beta < 8$, and $0 < \alpha + \beta < 8$. Thus, the number of all possible values of (α,β) is $7 + 6 + \cdots + 2 + 1 = 28$. However, similar to the case of BRIE, α and β should also obey the following rule pointed out in [19]: $\alpha \neq 1,7$ or $\alpha + \beta \neq 1,7$. If this rule is not satisfied, then there only exist 1-bit circular rotation operations, since Rotate $X_i^{p,1}$ = Rotate $X_i^{p,7}$ and Rotate $Y_j^{q,1}$ = Rotate $Y_j^{q,7}$. Generally speaking, 1-bit circular rotations are not good enough to effectively encrypt the plain image, and some visual information may leak from the cipher image. When $(\alpha,\beta) = (1,6)$, $x(0) = 33578/2^{17} \approx 0.2562$, $\mu = 129518/2^{15} \approx 3.9526$, the encryption results of two plain images, House and Cameraman, are shown in Figure 3. It can be seen that the visual information contained in the cipher images is so much (even more than that in Figure 2) that the plain images can be obviously guessed. Excluding the three values of (α,β) that violate the above rule, $(1,0)$, $(1,6)$, $(7,0)$, the number of all "good" values of (α,β) is only 25 $(= 28 - 3)$.

3.3. Low practical security against brute-force attacks

In [1, 2], it was claimed that the complexity of TDCEA against brute-force attack is $O(2^{17MN/8})$ since $17MN/8$ secret bits are used in the encryption/decryption procedures. However, this statement is not true due to the following reason: all $17MN/8$ bits are uniquely determined by the initial condition $x(0)$ and the control parameter μ of the logistic map (1), which have only 34 secret bits. Moreover, not all values of μ can ensure the chaoticity of the logistic map [27], so we can assure that the number of possible different chaotic bit sequences is smaller than 2^{34}.

Considering that the computational complexity of TDCEA is $O(MN)$, that is, $49MN$ operations of all kinds [1, Section 2.5], and the number of all possible values of (α,β) is 25, the total complexity against the brute-force attack is $O(2^{34} \cdot 25 \cdot 49MN) \approx O(2^{44}MN)$. For a typical image of size 256×256, the complexity is about $O(2^{60})$, which is much smaller than $O(2^{17MN/8}) = O(2^{139264})$, the complexity claimed in [1, 2]. Obviously, the security of TDCEA against brute-force attacks was overestimated too much in [1, 2].

4. KNOWN-PLAINTEXT ATTACKS

The known-plaintext attack is the attack of reconstructing the secret key or its equivalent with some known plaintexts and their corresponding ciphertexts, which is practical and occurs more and more frequently in today's networked world [28]. Although it was claimed that TDCEA can efficiently resist this kind of attacks [1, Section 2.6], we propose two different known-plaintext attacks in this section to effectively break TDCEA. One attack requires a few number of known plain texts, and another requires only one.

(a) (b)

Figure 3: Two natural images, House and Cameraman, encrypted by TDCEA, when $(\alpha,\beta) = (1,6)$, $x(0) = 33578/2^{17} \approx 0.2562$, and $\mu = 129518/2^{15} \approx 3.9526$. (a) Encrypted House. (b) Encrypted Cameraman.

4.1. Known-plaintext attack 1: getting permutation matrices as an equivalent key

The insecurity of BRIE against known-/chosen-plaintext attacks is caused by the fact that the ROLR operation is actually composed of secret permutations of all 8 bits of each pixel value. As shown in [25], all permutation-only ciphers are not secure against known-/chosen-plaintext attacks. Apparently, TDCEA falls into the category of permutation-only ciphers, since the circulation rotations are actually secret permutations of all 64 bits of each 8-pixel block. As a result, if an attacker knows (or chooses) a number of plain blocks and cipher blocks at the same position, k, it is possible for him to partially (or even completely) reconstruct the bit permutation by comparing M_k and M'_k. This is the basic principle of the first type of known-/chosen-plaintext attacks to be discussed below.

Apparently, for the kth pixel block $f^{(8)}(k)$ and its cipher block $f'^{(8)}(k)$, the encryption transformation can be represented by an 8×8 permutation matrix $W_k = [W_k(i,j)]_{i=0,j=0}^{7,7}$, where $W_k(i,j) = (i',j')$ denotes the secret position of the plain bit $M_k(i,j)$ in M'_k. Since there are $MN/8$ different blocks, the encryption of f can be represented by $MN/8$ permutation matrices: $\{W_k\}_{k=0}^{MN/8-1}$. Once the attacker gets the $MN/8$ permutation matrices and their inverses, $\{W_k^{-1}\}_{k=0}^{MN/8-1}$, he can use these matrices as an equivalent key to decrypt any cipher image encrypted with the same key.

In [25], a general algorithm was proposed for deriving the secret permutations (i.e., the permutation matrices) from a number of known plain images and their corresponding cipher images. This algorithm depends on the fact that the secret permutations do not change the values of the permuted elements. As a result, one can compare the values of the elements of the plain images and the cipher images to reveal the secret permutations. Here, we show how to optimally realize the general algorithm for TDCEA and discuss the breaking performance.

Given n known plain images $f_0 \sim f_{n-1}$ and their corresponding cipher images $f'_0 \sim f'_{n-1}$, denoting the kth 8×8 bit matrix of the lth plain image and cipher image by $M_{l,k} = [M_{l,k}(i,j)]_{i=0,j=0}^{7,7}$, $M'_{l,k} = [M'_{l,k}(i,j)]_{i=0,j=0}^{7,7}$, respectively, the algorithm of deriving the permutation matrix W_k is described as follows.

(i) *Step (1a)*. Calculate a generalized bit matrix $\widetilde{M}_k = [\widetilde{M}_k(i,j)]_{i=0,j=0}^{7,7}$, where $\widetilde{M}_k(i,j) = \sum_{l=0}^{n-1} M_{l,k}(i,j) \cdot 2^l$. Apparently, $\widetilde{M}_k(i,j)$ is an n-bit integer.

Note. When n is larger than the word length of the longest integer (which is 32 or 64 for most computers), it may be impossible to store $\widetilde{M}_k(i,j)$ as a normal integer in a computer. In this case, one has to divide $\widetilde{M}_k(i,j)$ into multiple short integers for storage and computation (i.e., to use long-integer techniques). Since the long-integer technique is easy for implementations and n is generally smaller than 32 in most attacking scenarios,[3] here we do not pay special attention to this issue.

(ii) *Step (1b)*. Calculate a generalized bit matrix $\widetilde{M}'_k = [\widetilde{M}'_k(i,j)]_{i=0,j=0}^{7,7}$, in the same way as Step (1a).

(iii) *Step (2)*. Get multivalued permutation matrix, $\widehat{W}_k = [\widehat{W}_k(i,j)]_{i=0,j=0}^{7,7}$, where $\widehat{W}_k(i,j) = \{(i',j') \mid \widetilde{M}_k(i,j) = \widetilde{M}'_k(i',j')\}$.

(iv) *Step (3)*. Derive an estimation of the permutation matrix W_k from $\widehat{W}_k$.

Apparently, if and only if each element of $\widehat{W}_k$ contains only one pixel position, that is, the measure of every element of $\widehat{W}_k$ is 1, one can uniquely get the permutation matrix W_k; otherwise, only an estimated version $\widetilde{W}_k$ can be derived. In other words, $\widetilde{W}_k = W_k$ holds if and only if the cardinality of $\widehat{\mathbb{W}}_k = \{\widehat{W}_k(0,0),\ldots,\widehat{W}_k(7,7)\}$ is 64, that is, $\#(\widehat{\mathbb{W}}_k) = 64$. When $\#(\widehat{\mathbb{W}}_k) = P < 64$, with n_i $(i = 1 \sim P)$ denoting the measure of the P different elements in $\widehat{W}_k$, one can easily deduce that there are $\prod_{i=1}^{P}(n_i!)$ possible estimations of W_k in total. Thus, the task of Step (3) is to determine one estimated permutation matrix from all possible $\prod_{i=1}^{P}(n_i!)$. Although many different methods can be used to realize Step (3), the following simple algorithm is enough in most cases to achieve an acceptable performance.

[3] As discussed below, the breaking performance is rather good when $n \leq 32$ (see Figure 5), so one can simply set $n = 32$ even when $n > 32$.

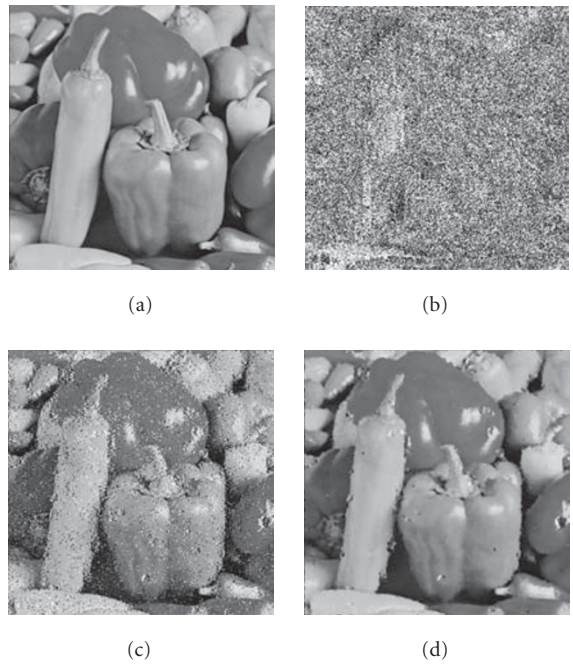


Figure 4: The image Peppers recovered by the first known-plaintext attack. (a) Peppers. (b) Encrypted Peppers. (c) Recovered Peppers via $\{\widetilde{W}_k\}_{k=0}^{MN/8-1}$. (d) Enhanced Peppers by a 3×3 median filter.

(i) Initialize all elements of an 8×8 flag matrix, $F_k = [F_k(i, j)]_{i=0,j=0}^{7,7}$, to zeros.
(ii) For $i = 0 \sim 7$ and $j = 0 \sim 7$, determine the value of $\widetilde{W}_k(i, j)$ as follows:
  (1) find the first position (i', j') satisfying $M_k(i, j) = M'_k(i', j')$ and $F_k(i', j') = 0$;
  (2) set $\widetilde{W}_k(i, j) = (i', j')$ and $F_k(i', j') = 1$.

Note that Step (2) is also incorporated into the above algorithm, which is very useful in reducing the total complexity.

Next, we will see how many known plain images are enough to achieve an acceptable breaking performance. Roughly, the larger the n, the less the $\prod_{i=1}^{P}(n_i!)$, the more accurate the estimated permutation matrix $\widetilde{W}_k$, and so the better the breaking performance will be. As a result, by estimating the mathematical expectation of n_i, one can conceptually derive a lower bound for n. To simplify the following analyses, we assume that each element in $M_{l,k}$ distributes uniformly over $\{0, 1\}$ and any two elements are independent of each other. Then, one can see that there are two types of elements in each $\widehat{W}_k(i, j)$:

(i) *the only real position*, which absolutely occurs;
(ii) *other fake positions*, each of which occurs in $\widehat{W}_k(i, j)$ with a probability of $1/2^n$, since any two bits in a bit matrix are identical with a probability of 1/2.

Thus, it follows that the average cardinality of $\widehat{W}_k(i, j)$ is $\overline{n}_i = 1+(64-1)/2^n = 1+63/2^n$, which approaches 1 exponentially as n increases. Generally speaking, when $1+63/2^n < 1.5$, that is, about half elements in $\widetilde{W}_k$ are correct, the decryption performance will be acceptable.[4] Solving this inequality, one has

$$n \geq 1 + \lceil \log_2 63 \rceil = 1 + \lceil 5.9773 \rceil = 7. \tag{8}$$

This theoretical result has been verified by experiments as shown in Figures 4 and 5. Note that the above analysis can also be derived from the general result given in [25]. Though the above result is deduced under the assumption that $\{M_{l,k}\}$ is an i.i.d. sequence, it can be qualitatively generalized to other distributions of $\{M_{l,k}\}$. Our experiments show that the above theoretical result essentially holds for most natural images.

For a randomly selected key, $(\alpha, \beta) = (2, 2)$, $x(0) = 33578/2^{17} \approx 0.2562$, $\mu = 129518/2^{15} \approx 3.9526$, a set of known plain images (all natural images) are randomly selected for testing. When $n = 8$, the plain image "Peppers" (Figure 4a) and its cipher image (Figure 4b) are used to verify the breaking performance based on $MN/8$ estimated permutation matrices, $\{\widetilde{W}_k\}_{k=0}^{MN/8-1}$. The recovered plain image is shown in Figure 4c. It is found that almost all visual information contained in the original plain image has been successfully recovered, though only $38012/65536 = 58\%$ of plain pixels are correct in value. With some noise reduction algorithms, one can further enhance the recovered plain image. One enhanced result with a 3×3 median filter is shown in Figure 4d.

[4] It is an empirical result drawn from our experiments, which can be qualitatively explained by the fact that human eyes have a good capability of rejecting noises in natural images.

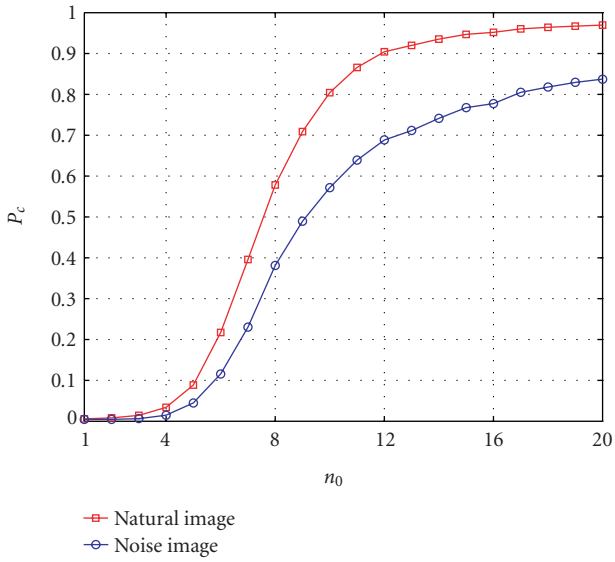


Figure 5: The percentage of correctly recovered pixels, P_c, with respect to the number of known plain images, n_0.

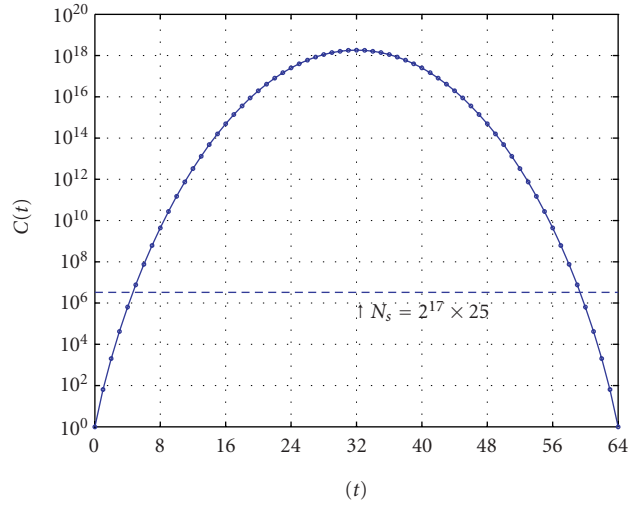


Figure 6: $C(t) = 64!/(t!(64-t)!)$ with respect to t.

Figure 5 shows the percentage of correctly recovered plain pixels with respect to n, the number of known plain images. One can see that the breaking performance is good when $n \geq 8$. Also, it is found that the breaking performance of the natural image is better than the noisy image under the same condition, which is attributed to the correlation existing in the natural image for decryption as discussed in [25]. It can also be observed that the slope of the two lines in Figure 5 are very flat when $n \geq 16$, this is also due to the correlation of the known images (e.g., the MSBs of adjacent pixels are the same with a high probability).

The complexity of this attack is rather small. For each block, the time complexity consumed in Step (1a) and Step (1b) is $O(2 \cdot 64 \cdot (n-1))$, and the average complexity in Step (2) is $O(64 \cdot 32)$, so the total attack complexity is only $O((2 \cdot 64 \cdot (n-1) + 64 \cdot 32) \cdot MN/8) = O(16(n+15)MN)$.

This known-plaintext attack has two disadvantages: (1) the number of required known plain images is somewhat large; (2) with n known plain images of size $M \times N$, this attack can only decrypt cipher images of size not greater than $M \times N$. In the following subsection, we will introduce another known-plaintext attack, by which we can get the secret keys with only one known plain image (but with a larger complexity).

4.2. Known-plaintext attack 2: getting the secret key from one known plain image

The known-plaintext attack introduced in this subsection is actually an optimized brute-force attack. By utilizing the correlation information existing between two consecutive chaotic states and the control parameter μ, the multiplicative search of the two subkeys $x(0)$ and μ can be reduced to be the additive search of two chaotic states $x(k)$ and $x(k+1)$. This can dramatically reduce the attack complexity. Also, since each guessed chaotic state can be verified by a few number of 8-pixel blocks, not by the whole known plain image, the attack complexity can be further reduced.

The basic idea of this attack is based on the following facts: (1) each permutation matrix W_k is uniquely determined by the current chaotic state $x(k)$ and the two subkeys α, β; (2) two consecutive chaotic states $x(k)$ and $x(k+1)$ satisfy $x(k+1) \approx \mu \cdot x(k) \cdot (1 - x(k))$. Once an attacker gets the right values of any two consecutive chaotic states, he can immediately get an estimation of μ, and then completely break TDCEA if α and β are also known.

To get the right value of a chaotic state $x(k)$ corresponding to the kth bit matrix M_k, one can use the permutation information existing in M_k and M'_k. When there are t 0-bits and $(64-t)$ 1-bits in M_k, one can calculate that the number of all possible values of M'_k is $C(t) = \binom{64}{t} = 64!/t!(64-t)!$. In comparison, the number of all possibilities of each permutation matrix is equal to the number of all possible values of the 3-tuple data $(x(k), \alpha, \beta)$, which is less than $N_s = 2^{17} \cdot 25$. When $5 \leq t \leq 59$, one has $C(t) \gg N_s$ (see Figure 6). This means that the probability that a wrong value of $(x(k), \alpha, \beta)$ coincides with W'_k is close to zero, that is, one can exhaustively search all possible values of $(x(k), \alpha, \beta)$ to find a few number of candidates of the right value. Apparently, such an exhaustive searching procedure is optimized when $t = 32$.

Carrying out the above procedure on two consecutive bit matrices, one can find some candidates of two consecutive chaotic states, $x(k) = 0.b(17k+0) \cdots b(17k+16)$ and $x(k+1) = 0.b(17k+17) \cdots b(17k+33)$. Then, an estimated value of the subkey μ can be derived as

$$\widetilde{\mu} = \frac{x(k+1)}{x(k) \cdot (1 - x(k))}. \tag{9}$$

Due to the quantization errors introduced in the finite-precision arithmetic, generally $x(k+1) \neq \mu \cdot x(k) \cdot (1 - x(k))$,

so $\widetilde{\mu} \neq \mu$. Fortunately, following the error analysis of $\widetilde{\mu}$ given in [22, Appendix], it has been shown that when $x(k+1) \geq 2^{-n}$ $(n = 1 \sim 17)$, $|\widetilde{\mu} - \mu| < 2^{n+3} \cdot 2^{-17}$. For example, when $x(k+1) \geq 2^{-1} = 0.5$, one can exhaustively search $2^4 = 16$ values in the neighborhood of $\widetilde{\mu}$ to find the right value of μ. To verify whether $\widetilde{\mu} = \mu$, one can iterate the logistic map from $x(k+1)$ until $x(MN/8-1)$ and then check the coincidence between each bit matrix M_i and M_i', $i = k+2, \ldots, MN/8-1$. Once a mismatch occurs, the current guessed value is discarded, and the next value is tested. To minimize the verification complexity, one can check only a number of chaotic states sufficiently far from $x(k+1)$ to eliminate most (if not all) wrong values of $\widetilde{\mu}$, and verify a few left ones by checking all chaotic states from $x(k+2)$ to $x(MN/8-1)$.

The proposed known-plaintext attack can be concretized step by step as follows.

(i) *Step (1).* Find the first two consecutive plain blocks $f^{(8)}(k)$ and $f^{(8)}(k+1)$, whose corresponding bit matrices M_k and M_{k+1} both have about 32 0-bits.

Note. Assuming that each bit in M_k distributes uniformly and independently, one can deduce that

$$P_s = \text{Prob}\left[|t-32| \leq s\right] = \frac{\sum_{i=32-s}^{32+s} \binom{64}{i}}{2^{64}}, \qquad (10)$$

where t is the number of nonzero elements of M_k and $0 \leq s \leq 32$. When $s = 4$, $P_s \approx 0.7396$, which is sufficiently large for an attacker to find valid plain-blocks within all the $MN/8$ blocks.

(ii) *Step (2).* Exhaustively search all possible values of $(x(k), \alpha, \beta)$ and record those coinciding with M_k and M_k'. Assume that m_1 candidates are recorded in total: $\{x_i(k), \alpha_i^*, \beta_i^*\}_{i=0}^{m_1-1}$.

(iii) *Step (3).* Search all possible values of $x(k+1)$ and all values of (α, β) in $\{\alpha_i^*, \beta_i^*\}_{i=0}^{m_1-1}$ and record those coinciding with M_{k+1} and M_{k+1}'. Assume that m_2 candidates are recorded in total: $\{x_j(k+1), \alpha_j^{**}, \beta_j^{**}\}_{j=0}^{m_2-1}$.

(iv) *Step (4).* For $i = 0 \sim m_1 - 1$ and $j = 0 \sim m_2 - 1$, do the following operations.

Step (4a). If $\alpha_i^* = \alpha_j^{**}$ and $\beta_i^* = \beta_j^{**}$, then calculate $\widetilde{\mu} = x_j(k+1)/x_i(k) \cdot (1 - x_i(k))$ and continue to execute Step (4b); otherwise, go to the next loop.

Step (4b). Assuming that $x_j(k+1) \geq 2^{-n}$, exhaustively search all possible 2^{n+3} values of μ within the neighborhood of $\widetilde{\mu}$. For each searched value, iterate the logistic map from $x_i(k+1)$ to $x_i(MN/8-1)$. If every chaotic state $x_i(l)$ and (α_i^*, β_i^*) agree with M_l and M_l' $(l = k+2 \sim MN/8-1)$, then the attack completes.

The time complexity of this attack can be calculated as follows.

Figure 7: The recovered plain image Peppers by the second known-plaintext attack.

(i) The average complexity of Step (2) is $2^{17} \cdot 25 \cdot (14 \cdot 8 + (1/2) \cdot 8 \cdot 8) < 2^{29}$.

(ii) The complexity of Step (3) is obviously less than that of Step (2).

(iii) The average number of exhaustive searching loops in Step (4) is $(m_1 \cdot m_2 \cdot C_x)$, where

$$C_x = \sum_{n=1}^{17} 2^{n+3} \cdot \text{Prob}\left[2^{-n} \leq x_j(k+1) < 2^{-(n-1)}\right], \qquad (11)$$

which is the mathematical expectation of the space size of the searching neighborhood of $\widetilde{\mu}$. Considering the computational complexity for each searching loop, the average complexity of Step (4) is of order $(m_1 \cdot m_2 \cdot C_x/2) \cdot 49MN$. Without loss of generality, assume that $x_j(k+1)$ distributes uniformly over the interval $[0,1]$, that is, $\text{Prob}[2^{-n} \leq x_j(k+1) < 2^{-(n-1)}] = 2^{-n}$. Thus, $C_x = \sum_{n=1}^{17} 2^{n+3} \cdot 2^{-n} = 2^3 \cdot 17 = 136$. Then, the average complexity becomes $O(833 m_1 m_2 MN/2)$. Since, in most cases, $MN \leq 4096 \cdot 4096 = 2^{24}$ and m_1, m_2 are generally very small, the complexity is generally not greater than $O(2^{36})$.

Combining the above results, one concludes that the total complexity is $O(2^{36})$, which is practically small even for a PC and much smaller than $O(2^{60})$, the complexity of the simple brute-force attack shown in Section 3.3.

Figure 7 shows an experimental result of the recovered plain image "Peppers," where the fifth and sixth pixel blocks are chosen to exhaustively search the secret key. As a result, all chaotic states from $x(5)$ are successfully derived and only $(5 \cdot 8 = 40)$ leading plain pixels at the left-bottom corner are not recovered correctly.

5. CHOSEN-PLAINTEXT ATTACKS

Chosen-plaintext attacks are enhanced (and generally stronger) versions of known-plaintext attacks, with some intentionally chosen plaintexts and their corresponding ciphertexts [28]. In these attacks, the two known-plaintext attacks introduced in the previous section can be significantly enhanced.

5.1. Chosen-plaintext attack 1: getting permutation matrices as an equivalent key

As discussed in Section 4.1, if $\#(\widetilde{\mathbb{W}}_k) = 64$, the permutation matrix W_k can be uniquely determined. Apparently, it is easy to ensure $\#(\widetilde{\mathbb{W}}_k) = 64$ by choosing the following six plain images for all $k = 0 \sim MN/8 - 1$, $i = 0 \sim 7$, $j = 0 \sim 7$:

$$\begin{aligned} f_0 &: M_{0,k}(i,j) = \left\lfloor \frac{(8i+j)}{32} \right\rfloor \bmod 2; \\ f_1 &: M_{1,k}(i,j) = \left\lfloor \frac{(8i+j)}{16} \right\rfloor \bmod 2; \\ f_2 &: M_{2,k}(i,j) = \left\lfloor \frac{(8i+j)}{8} \right\rfloor \bmod 2; \\ f_3 &: M_{3,k}(i,j) = \left\lfloor \frac{(8i+j)}{4} \right\rfloor \bmod 2; \\ f_4 &: M_{4,k}(i,j) = \left\lfloor \frac{(8i+j)}{2} \right\rfloor \bmod 2; \\ f_5 &: M_{5,k}(i,j) = (8i+j) \bmod 2. \end{aligned} \tag{12}$$

With the above six chosen plain images, $\#(\widetilde{\mathbb{W}}_k) = 64$ holds so all $MN/8$ permutation matrices can be uniquely determined, which can then be used to decrypt any cipher image of size not greater than MN.

The time complexity of such an attack is of the same order as the known-plaintext attack with $n = 6$ known plain images, that is, $O(16(6+15)MN) = O(336MN)$.

In fact, due to a special weakness of TDCEA, even two chosen plain images are enough to completely reconstruct each 8×8 permutation matrix. Recalling the encryption procedure of TDCEA, one can see that 2D secret rotations are merely a simple combination of 1D rotations in two directions: 8 horizontal rotations followed by 8 vertical rotations. Such a property makes the division of the 2D secret rotations possible in chosen-plaintext attacks with only two plain images. In cryptanalysis, we call such attacks *divide-and-conquer* (DAC) attacks. The DAC chosen-plaintext attack can be described as follows.

(i) *Break the 8 vertical secret rotations*. Choose a plain image f_0 as follows: for all $k = 0 \sim MN/8 - 1$, $f_0^{(8)}(k) = \{255, 0, 0, 0, 0, 0, 0, 0\}$, that is,

$$M_{0,k} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{13}$$

It is obvious that the 8 horizontal secret rotations have no influence on the above plain image. That is, the 2D TDCEA is reduced to the 1D BRIE in the vertical direction. Since each column of $M_{0,k}$ has only one 1-bit, by comparing $M_{0,k}$ and $M'_{0,k}$ one can uniquely get 8 values, $s_k(j)$ ($j = 0 \sim 7$), which satisfy $M'_{0,k} = \text{Rotate } Y_j^{0,s_k(j)}(M_{0,k})$ and serves as the equivalent rotation parameter of the jth column.

(ii) *Break the 8 horizontal secret rotations*. Choose a plain image f_1 as follows: for all $k = 0 \sim MN/8-1$, $f_1^{(8)}(k) = \{1, 1, 1, 1, 1, 1, 1, 1\}$, that is,

$$M_{1,k} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{14}$$

Since the 8 vertical secret rotations have been obtained via f_0, one can remove all the 8 vertical rotations from $M'_{1,k}$ to get the intermediate bit matrix $M^*_{1,k}$. Then, by comparing $M^*_{1,k}$ and $M_{1,k}$, one can similarly get another 8 values, $r_k(i)$ ($i = 0 \sim 7$), where $M^*_{1,k} = \text{Rotate } X_i^{0,r_k(i)}(M_{1,k})$. Here, $r_k(0) \sim r_k(7)$ are the equivalent rotation parameters of the ith line.

Apparently, after revealing the horizontal and vertical secret rotations, the permutation matrix W_k can be immediately reconstructed by simply combining the 16 rotations. In this case, the time complexity is only $O((4+1+4+8)MN) = O(17MN)$.

5.2. Chosen-plaintext attack 2: getting the secret key

In the first chosen-plaintext attack, one can get 16 values, $s_k(0) \sim s_k(7)$ and $r_k(0) \sim r_k(7)$, for each pixel block $f^{(8)}(k)$. Based on the 16 values, the second known-plaintext attack discussed in Section 4.2 can be dramatically enhanced in most cases by introducing a much more effective way of deriving the 17 secret bits, $b(17k+0) \sim b(17k+16)$, of the chaotic state $x(k)$.

To simplify the following discussions, create a new vector, $rs_k(i)$ ($i = 0 \sim 15$), which satisfies that for all $i = 0 \sim 7$, $rs_k(i) = r_k(i)$ and for all $i = 8 \sim 15$, $rs_k(i) = s_k(i-8)$.

Recalling the encryption procedure of TDCEA, it is obvious that the 16 values $\{rs_k(i)\}_{i=0}^{15}$ have a deterministic relation with the 17 secret bits $b(17k+0) \sim b(17k+16)$. Such a relation can be used to facilitate an exhaustive search of the 17 secret bits, that is, the search of the kth chaotic state $x(k) = 0.b(17k+0) \cdots b(17k+16)$.

Considering the fact that $\text{Rotate } X_i^{0,r} = \text{Rotate } X_i^{1,8-r}$, $\text{Rotate } Y_j^{0,s} = \text{Rotate } Y_j^{1,8-s}$, one can see that for all $i = 0 \sim 15$, $k = 0 \sim MN/8 - 1$, $rs_k(i)$ must be a value in the set $\mathbb{S} = \{\alpha, \alpha+\beta, 8-\alpha, 8-(\alpha+\beta)\}$.

For each guessed value $(\widetilde{\alpha}, \widetilde{\beta})$, one can determine 16 bits, denoted by $\widetilde{b}(17k+1) \sim \widetilde{b}(17k+16)$, as estimations of $b(17k+1) \sim b(17k+16)$, as follows: for all $i = 1 \sim 16$,

$$\widetilde{b}(17k+i) = \begin{cases} 0, & rs_k(i-1) \in \{\widetilde{\alpha}, 8-\widetilde{\alpha}\}, \\ 1, & rs_k(i-1) \in \{\widetilde{\alpha}+\widetilde{\beta}, 8-(\widetilde{\alpha}+\widetilde{\beta})\}. \end{cases} \tag{15}$$

Note that the above equation is invalid when $\widetilde{\alpha} = \widetilde{\alpha} + \widetilde{\beta}$ or $\widetilde{\alpha} = 8 - (\widetilde{\alpha} + \widetilde{\beta})$, that is, $\widetilde{\beta} = 0$ or $2\widetilde{\alpha} + \widetilde{\beta} = 8$. Similarly, one has another equation for estimating the values of $b(17k + 0) \sim b(17k + 15)$: for all $i = 0 \sim 15$,

$$\widetilde{b}(17k + i) = \begin{cases} 0, & rs_k(i) \in \{\widetilde{\alpha}, \widetilde{\alpha} + \widetilde{\beta}\}, \\ 1, & rs_k(i) \in \{8 - \widetilde{\alpha}, 8 - (\widetilde{\alpha} + \widetilde{\beta})\}. \end{cases} \tag{16}$$

The above equation is invalid when $\widetilde{\alpha} = 4$, $\widetilde{\alpha} + \widetilde{\beta} = 4$ or $2\widetilde{\alpha} + \widetilde{\beta} = 8$.

According to how much information one can get from $\{rs_k(i)\}_{i=0}^{15}$, all values of $(\widetilde{\alpha}, \widetilde{\beta})$ can be divided into the following classes in the chosen-plaintext attack.

(C1) $\widetilde{\alpha} \neq 4$, $\widetilde{\alpha} + \widetilde{\beta} \neq 4$, $\widetilde{\beta} \neq 0$ *and* $2\widetilde{\alpha} + \widetilde{\beta} \neq 8$: $\widetilde{b}(17k + 1) \sim \widetilde{b}(17k + 16)$ and $\widetilde{b}(17k + 0) \sim \widetilde{b}(17k + 15)$ can be uniquely determined by (15) and (16), respectively, so all the 17 bits, $\widetilde{b}(17k+0) \sim \widetilde{b}(17k+16)$, can be uniquely recovered.

There are 12 *(C1)*-values of $(\widetilde{\alpha}, \widetilde{\beta})$, as follows: $(1,1)$, $(1,2)$, $(1,4)$, $(1,5)$, $(2,1)$, $(2,3)$, $(2,5)$, $(3,3)$, $(3,4)$, $(5,1)$, $(5,2)$, $(6,1)$.

(C2) $4 \in \{\widetilde{\alpha}, \widetilde{\alpha} + \widetilde{\beta}\}$ *and* $\widetilde{\beta} \neq 0$ *(which ensures* $2\widetilde{\alpha} + \widetilde{\beta} \neq 8$*)*: $\widetilde{b}(17k + 1) \sim \widetilde{b}(17k + 16)$ can be uniquely determined by (15), but $\widetilde{b}(17k + 0)$ has to be guessed.[5]

There are 6 *(C2)*-values of $(\widetilde{\alpha}, \widetilde{\beta})$, as follows: $(1,3)$, $(2,2)$, $(3,1)$, $(4,1)$, $(4,2)$, $(4,3)$.

(C3) $\widetilde{\alpha} \neq 4$ *and* $\widetilde{\beta} = 0$ *(which ensures* $2\widetilde{\alpha} + \widetilde{\beta} \neq 8$*)*: $\widetilde{b}(17k + 0) \sim \widetilde{b}(17k + 15)$ can be uniquely determined by (16), but $\widetilde{b}(17k + 16)$ has to be guessed.

There are 6 *(C3)*-values of $(\widetilde{\alpha}, \widetilde{\beta})$, as follows: $(1,0)$, $(2,0)$, $(3,0)$, $(5,0)$, $(6,0)$, $(7,0)$.

(C4) $2\widetilde{\alpha} + \widetilde{\beta} = 8$: all the 17 bits has to be exhaustively guessed, as in the second known-plaintext attack discussed in Section 4.2.

There are 4 *(C4)*-values of $(\widetilde{\alpha}, \widetilde{\beta})$, as follows: $(1,6)$, $(2,4)$, $(3,2)$, $(4,0)$.

The above four different cases correspond to different values of $\#(\mathbb{S})$ as follows:

(i) $\#(\mathbb{S}) = 4$: $(\widetilde{\alpha}, \widetilde{\beta})$ is one of the 12 *(C1)*-values;

(ii) $\#(\mathbb{S}) = 3$: $(\widetilde{\alpha}, \widetilde{\beta})$ is one of the 6 *(C2)*-values;

(iii) $\#(\mathbb{S}) = 2$: $(\widetilde{\alpha}, \widetilde{\beta})$ is one of the 6 *(C3)*-values and the following *(C4)*-values: $\{(1,6), (2,4), (3,2)\}$;

(iv) $\#(\mathbb{S}) = 1$: $(\widetilde{\alpha}, \widetilde{\beta}) = (4,0)$ (a *(C4)*-value).

Since one can guess the value of $\#(\mathbb{S})$ by observing the cardinality of the set $\{rs_k(0), \ldots, rs_k(15)\} \subseteq \mathbb{S}$, it is possible to search (α, β) in part of all possible values to reduce the attack complexity. Apparently, the success probability of such a guess is $P_e = \text{Prob}[\mathbb{S} = \{rs_k(0), \ldots, rs_k(15)\}]$. Since the theoretical deduction of P_e is rather difficult, experiments are performed to test all 2^{17} possible values of $b(17k + 0) \sim b(17k + 16)$. It results in that $P_e = 122684/2^{17} \approx 0.936$, which is sufficiently large. Note that it is easy to further increase the success probability of the guess, by observing $n > 1$ blocks at the same time. In doing so, the success probability will be *greater* than $P_e^{(n)} = 1 - (1 - P_e)^n$ under the assumption that the chaotic bits for different blocks distribute uniformly and independently. As n increases, $P_e^{(n)}$ will approach 1 exponentially. In real attacks, even $n = 2$ is enough in almost all cases, since $P_e^{(2)} \approx 0.996$. If all guessed values determined by $\#(\{rs_k(0), \ldots, rs_k(15)\})$ fail to pass the verification, it means that the rare event $\{rs_k(0), \ldots, rs_k(15)\} \subset \mathbb{S}$ occurs.[6] In this case, one has to continue to exhaustively search all other values of (α, β).

When the real value of (α, β) belongs to *(C1)*, *(C2)*, *(C3)* classes, the complexity of the chosen-plaintext attack will be much smaller than the complexity of its known-plaintext counterpart, due to the following reasons:

(i) the exhaustive searching procedure for the 17 bits of each chaotic state is simplified to be a deterministic calculation procedure dominated by (15) and/or (16);

(ii) the number of guessed values of (α, β) is reduced from 28 to 12 for *(C1)*, 6 for *(C2)* and *(C3)*;

(iii) some values of (α, β) can be verified by checking whether or not $\{rs_k(0), \ldots, rs_k(15)\} \subseteq \{\widetilde{\alpha}, \widetilde{\alpha} + \widetilde{\beta}, 8 - \widetilde{\alpha}, 8 - (\widetilde{\alpha} + \widetilde{\beta})\}$;

(iv) one can intentionally choose the second chaotic state to ensure $x(k + 1) \geq 0.5$, that is, $b(17(k + 1) + 0) = 1$, so as to reduce C_x, the average searching complexity of μ, from 136 to $2^{1+3} = 16$;

(v) the exhaustive search of μ can be validated by just comparing the calculated chaotic state with the bits derived by (15) and/or (16).

When the real value of (α, β) belongs to *(C4)* class, the average complexity of the chosen-plaintext attack is also smaller than the one of its known-plaintext counterpart, since the value of (α, β) can be immediately determined[7] with a sufficiently high probability, $P_e^{(n)} \approx 1$, that is, only when the rare event $\{rs_k(0), \ldots, rs_k(15)\} \subset \mathbb{S}$ occurs, one needs to exhaustively search the value of (α, β).

[5] Note that $\widetilde{b}(17k + 0)$ can be uniquely determined in the following two subcases: (a) when $\widetilde{\alpha} = 4$ and $\widetilde{b}(17k + 1) = 1$, one can uniquely determine $\widetilde{b}(17k + 0)$ by (16) since $\widetilde{\alpha} + \widetilde{\beta} \neq 4$; (b) when $\widetilde{\alpha} \neq 4$ and $\widetilde{b}(17k + 1) = 0$, one can also uniquely determine $\widetilde{b}(17k + 0)$ by (16). The two subcases occur with a probability of 0.5 when $\{b(i)\}$ distributes uniformly over $\{0, 1\}$.

[6] Note that the occurrence probability is not zero, though it is very close to zero when n is sufficiently large.

[7] If $\#(\mathbb{S}) = 1$, then $(\alpha, \beta) \equiv (4,0)$; otherwise, one can determine the value of (α, β) quickly by checking the following three candidates: $(1,6), (2,4), (3,2)$.

6. CHOSEN-CIPHERTEXT ATTACKS

Chosen-ciphertext attacks are mirror versions of chosen-plaintext attacks, in which a cryptanalyst attempts to determine the secret key from knowledge of plaintexts that correspond to ciphertexts chosen by the attacker [28]. For TDCEA, due to the symmetry of the encryption and decryption procedures, one can carry out chosen-ciphertext attacks, in very much the same way as the chosen-plaintext attacks discussed in Section 5.

7. CONCLUSIONS

In this paper, the security of the recently proposed encryption scheme for multimedia transmission, called TDCEA [1, 2], has been analyzed carefully. Some defects existing in TDCEA have been found and diagnosed. Two methods of known-plaintext attacks and their chosen-plaintext attack counterparts have been proposed to break the scheme. In addition, a chosen-ciphertext attack has been mentioned briefly. Both theoretical and experimental analyses have been given to demonstrate the defects of TDCEA and to verify the feasibility of the proposed known-plaintext attacks. In conclusion, TDCEA is not suggested for applications that require a high level of security level.

ACKNOWLEDGMENTS

This research was partially supported by the National Natural Science Foundation, China, under Grants no. 60373041, no. 90104034, and no. 60202002, and by the Applied R&D Centers of the City University of Hong Kong under Grants no. 9410011 and no. 9620004.

REFERENCES

[1] H.-C. Chen, J.-I. Guo, L.-C. Huang, and J.-C. Yen, "Design and realization of a new signal security system for multimedia data transmission," *EURASIP J. Appl. Signal Process.*, vol. 2003, no. 13, pp. 1291–1305, 2003.

[2] J.-C. Yen and J.-I. Guo, "Design of a new signal security system," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 4, pp. 121–124, Scottsdale, Ariz, USA, May 2002.

[3] J.-C. Yen and J.-I. Guo, "A new image encryption algorithm and its VLSI architecture," in *Proc. IEEE Workshop on Signal Processing Systems (SiPS '99)*, pp. 430–437, Taipei, Taiwan, October 1999.

[4] J.-C. Yen and J.-I. Guo, "A new MPEG/encryption system and its VLSI architecture," in *Proc. International Symposium on Communications*, pp. 215–219, Kaohsiung, Taiwan, November 1999.

[5] K.-L. Chung and L.-C. Chang, "Large encrypting binary images with higher security," *Pattern Recognition Letters*, vol. 19, no. 5-6, pp. 461–468, 1998.

[6] N. Bourbakis and C. Alexopoulos, "Picture data encryption using scan patterns," *Pattern Recognition*, vol. 25, no. 6, pp. 567–581, 1992.

[7] C. Alexopoulos, N. Bourbakis, and N. Ioannou, "Image encryption method using a class of fractals," *J. of Electronic Imaging*, vol. 4, no. 3, pp. 251–259, 1995.

[8] L. Tang, "Methods for encrypting and decrypting MPEG video data efficiently," in *Proc. 4th ACM International Conference on Multimedia*, pp. 219–229, Boston, Mass, USA, November 1996.

[9] H. C. H. Cheng, "Partial encryption for image and video communication," M.S. thesis, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, 1998.

[10] L. Qiao, *Multimedia security and copyright protection*, Ph.D dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Ill, USA, 1998.

[11] S. U. Shin, K. S. Sim, and K. H. Rhee, "A secrecy scheme for MPEG video data using the joint of compression and encryption," in *Proc. 2nd International Information Security Workshop (ISW '99)*, vol. 1729 of *Lecture Notes in Computer Science*, pp. 191–201, Kuala Lumpur, Malaysia, November 1999.

[12] J.-C. Yeo and J.-I. Guo, "Efficient hierarchical chaotic image encryption algorithm and its VLSI realisation," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 147, no. 2, pp. 167–175, 2000.

[13] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 118–129, 2003.

[14] B. Bhargava, C. Shi, and S.-Y. Wang, "MPEG video encryption algorithms," *Multimedia Tools and Applications*, vol. 24, no. 1, pp. 57–79, 2004.

[15] J.-K. Jan and Y.-M. Tseng, "On the security of image encryption method," *Inform. Process. Lett.*, vol. 60, no. 5, pp. 261–265, 1996.

[16] L. Qiao, K. Nahrstedt, and M.-C. Tam, "Is MPEG encryption by using random list instead of zigzag order secure?," in *Proc. IEEE International Symposium on Consumer Electronics (ISCE '97)*, pp. 226–229, Singapore, December 1997.

[17] L. Qiao and K. Narhstedt, "Comparison of MPEG encryption algorithms," *Comput. Graphics*, vol. 22, no. 4, pp. 437–448, 1998.

[18] T. Uehara and R. Safavi-Naini, "Chosen DCT coefficients attack on MPEG encryption schemes," in *Proc. IEEE Pacific Rim Conference on Multimedia (PCM '00)*, pp. 316–319, Sydney, Australia, December 2000.

[19] S. Li and X. Zheng, "On the security of an image encryption method," in *Proc. IEEE International Conference on Image Processing (ICIP '00)*, vol. 2, pp. 925–928, Rochester, NY, USA, September 2002, http://www.hooklee.com/pub.html.

[20] S. Li and X. Zheng, "Cryptanalysis of a chaotic image encryption method," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 2, pp. 708–711, Phoenix-Scottsdale, Ariz, USA, May 2002, http://www.hooklee.com/pub.html.

[21] C.-C. Chang and T.-X. Yu, "Cryptanalysis of an encryption scheme for binary images," *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1847–1852, 2002.

[22] C. Li, S. Li, D. Zhang, and G. Chen, "Cryptanalysis of a chaotic neural network based multimedia encryption scheme," in *Advances in Multimedia Information Processing, Proc. 5th Pacific Rim Conference on Multimedia (PCM '04), part III*, vol. 3333 of *Lecture Notes in Computer Science*, pp. 418–425, Springer, Tokyo, Japan, November-December 2004.

[23] X.-Y. Zhao, G. Chen, Zhang D., X.-H. Wang, and G.-C. Dong, "Decryption of pure-position permutation algorithms," *J. Zhejiang Univ. SCIENCE*, vol. 5, no. 7, pp. 803–809, 2004.

[24] S. Li, C. Li, G. Chen, and X. Mou, "Cryptanalysis of the RCES/RSES image encryption scheme," 2004, Cryptology ePrint Archive: Report 2004/376, http://eprint.iacr.org/2004/376.

[25] S. Li, C. Li, G. Chen, D. Zhang, and N. G. Bourbakis, "A general cryptanalysis of permutation-only multimedia encryption algorithms," 2004, Cryptology ePrint Archive: Report 2004/374, http://eprint.iacr.org/2004/374.

[26] S. Li, G. Chen, and X. Zheng, "Chaos-based encryption for digital images and videos," in *Multimedia Security Handbook*, B. Furht and D. Kirovski, Eds., chapter 4, CRC Press, LLC, Boca Raton, Fla, USA, 2004, http://www.hooklee.com/pub.html.

[27] H. Bai-Lin, *Starting with Parabolas: An Introduction to Chaotic Dynamics*, Shanghai Scientific and Technological Education Publishing House, Shanghai, China, 1993.

[28] B. Schneier, *Applied Cryptography–Protocols, Algorithms, and Souce Code in C*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1996.

Chengqing Li received his B.S. degree in pure mathematics from the XiangTan University, Xiangtan, Hunan, China, in 2002. He is currently pursuing his M.S. degree in Zhejiang University, Hangzhou, Zhejiang, China. His current research interests include image/video encryption, image processing, and watermarking.

Shujun Li received his B.S. degree in information science and engineering, and his Ph.D. degree in information and communication engineering, both from the Xi'an Jiaotong University, Xi'an, China, in 1997 and 2003, respectively. Currently he is a Senior Research Assistant with the Center for Chaos Control and Synchronization, City University of Hong Kong, Hong Kong. His current research interests are in the fields of chaotic encryption, image and video encryption, statistical aspects of digital chaotic systems, secure human-computer identification, and visual/graphical passwords.

Guanrong Chen received the M.S. degree in computer science from Zhongshan University, China and the Ph.D. degree in applied mathematics from Texas A&M University, USA. Currently he is a Chair Professor and the Founding Director of the Centre for Chaos Control and Synchronization at the City University of Hong Kong. He has been a Fellow of the IEEE since 1996 for his fundamental contributions to the theory and applications of chaos control and bifurcation analysis. Professor Chen has (co)authored 15 research monographs and advanced textbooks, more than 300 SCI journal papers, and about 200 refereed conference papers, published since 1981 in the fields of nonlinear system dynamics and controls. He is serving as an Editor for 8 international journals including IEEE Transactions on Circuits and Systems, IEEE Transactions on Automatic Control, and International Journal of Bifurcation and Chaos, and received three best journal paper awards. He is an Honorary Professor of the Central Queensland University, Australia, and of more than ten Universities in China.

Gang Chen received his Bachelor of Science degree from Anqing Teachers College in 1983 and his Ph.D. degree from the Department of Applied Mathematics, Zhejiang University in 1994. Between 1994 and 1996, he was a Postdoctoral Researcher in electrical engineering at Zhejiang University. From 1997 to 1999, he worked as a Visiting Researcher in the Institute of Mathematics at the Chinese University of Hong Kong and the Department of Electronic and Information Engineering at The Hong Kong Polytechnic University. Since 2001, he has been a Professor at Zhejiang University. He has been the Director of the Institute of DSP and Software Techniques at Ningbo University since 2002. His research interests include applied mathematics, image processing, fractal geometry, wavelet analysis, and computer graphics. Dr. Chen has coauthored three books and coedited five technical proceedings. He has published more than 80 technical papers.

Lei Hu received the B.S. degree and the M.S. degree from Peking University, Beijing, China, in 1988 and 1991, respectively, and received the Ph.D. degree from the Institute of System Sciences, Chinese Academy of Sciences, Beijing, China, in 1994. Since 2002 he has been a Professor at the Graduate School of the Chinese Academy of Sciences. His research interests include cryptography and network security.