On the security of an image encryption scheme

Chengqing Li [a,*], Shujun Li [b,*], Juana Nunez [c],
Gonzalo Alvarez [c] and Guanrong Chen [a]

[a] *Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong SAR, China*

[b] *Faculty of Electrical and Computer Engineering, FernUniversität in Hagen, 58084 Hagen, Germany*

[c] *Instituto de Física Aplicada, Consejo Superior de Investigaciones Científicas, Serrano 144, 28006 Madrid, Spain*

Abstract

Recently, a new image encryption scheme using chaotic Logistic map was proposed. This paper studies the security properties of the scheme, and finds the following problems: 1) there is a number of secret keys that fail to serve as eligible secure ones; 2) one sub-key K_{10} may be guessed by observing only the cipher-image of a special plain-image; 3) there exist some potential insecure properties; 4) sub-keys $K_4 \sim K_{10}$ can be recovered with at most 64 pairs of differential chosen plain-images, being the attack performance especially good when K_{10} is not too large.

Key words: chaos, cryptanalysis, encryption, differential attack

1 Introduction

Owing to the rapid development of multimedia and network technologies, the transmission of multimedia data over networks occurs more and more frequently. Therefore, the secure protection of multimedia data, especially digital images and videos, is urgently needed. However, the traditional text ciphers, like DES and AES, fail to agree well with the properties and requirements of multimedia application, such as the bulky size and strong redundancy in

* Corresponding authors: Chengqing Li (cqli@ee.cityu.edu.hk), Shujun Li (http://www.hooklee.com).

the uncompressed multimedia data, the high encryption speed for real time process and the feasibility of cascade for the whole system. To meet the challenges, a great number of encryption schemes were proposed in the past two decades [1–7]. Meanwhile, security analysis on the proposed schemes have also been developed, and some of them have been found to be insecure to different extents, from the point of view of cryptography [8–12]. A more comprehensive survey of the state-of-the-art of this topic can be found in [13–15].

Since 2003, Pareek et al. have proposed three different encryption schemes based on one or more one-dimensional chaotic maps [16–18]. The two schemes proposed in [16] and [17] have been cryptanalyzed successfully in [19] and [20], respectively. In [18], a scheme based on Logistic map was proposed specially for image encryption. The present paper focuses on the security analysis of such scheme, and finds the following problems:

(1) There are some different types of security problems with the secret key, and each sub-key at least suffers from one of them;
(2) The main encryption functions have potential security holes, and the histogram of sub-images of a cipher image is not uniform enough;
(3) The scheme is insecure against a differential chosen-plaintext attack in the sense that only 64 pairs of chosen images are enough to break the scheme. The attack is especially feasible when K_{10} is not too large.

The rest of the paper is organized as follows. The next section introduces Pareek et al.'s scheme briefly. Section 3 gives detailed cryptanalysis of the scheme. The last section concludes the paper.

2 Pareek et al.'s scheme

The scheme scans the plain-image in reverse scan order, and encrypts it block by block, where each block contains 16 consecutive pixels. Without loss of generality, assume that the size of the plain-image is $M \times N$ (height$\times$width), and that MN can be exactly divided by 16. Then the plain-image $\{I(l)\}_{l=0}^{MN-1}$ has $MN/16$ blocks, namely, $\{I^{(16)}(0), \ldots, I^{(16)}(k), \ldots, I^{(16)}(MN/16-1)\}$, where $I^{(16)}(k)$ =$\{I(16k+0), \ldots,\ I(16k+i), \ldots, I(16k+15)\}$.

The secret key of the encryption scheme under study is an 80-bit integer and can be represented as $K = K_1 \cdots K_{10}$, where each sub-key $K_i \in \{0, \ldots, 255\}$. Two chaotic maps are used in the encryption scheme, both of which are Logistic maps defined by the following equation:

$$f(x) = \mu \cdot x \cdot (1-x), \tag{1}$$

where μ is the control parameter and it is fixed as 3.9999 throughout the scheme.

For the k-th pixel-block $I^{(16)}(k)$, the scheme can be described as follows.

Step 1. Generating the initial condition of the first Logistic map. The initial condition of the first Logistic map, X_0, is determined by the six sub-keys $K_4, \ldots, K_9$ as follows:

$$X_0 = \left(\frac{\sum_{i=4}^{6} K_i \cdot 2^{8(i-4)}}{2^{24}} + \frac{\sum_{i=7}^{9}((K_i \bmod 16) + \lfloor K_i/16 \rfloor)}{96} \right) \bmod 1. \tag{2}$$

Step 2. Generating the initial condition of the second Logistic map. The initial condition of the second Logistic map, Y_0, is determined by the chaotic state of the first Logistic map as follows. Iterate the first Logistic map to extract 24 chaotic states $\{\hat{X}_j\}_{j=1}^{24}$, by discarding the states not belonging to the interval $[0.1, 0.9)$ and then generate 24 integers $\{P_j\}_{j=1}^{24}$, where $P_j = \lfloor 24(\hat{X}_j - 0.1)/0.8 \rfloor + 1$.[1] Then, calculate $B_2 = \sum_{i=1}^{3} K_i \cdot 2^{8(i-1)}$ and set

$$Y_0 = \left(\frac{B_2 + \sum_{j=1}^{24} B_2[P_j] \cdot 2^{k-1}}{2^{24}} \right) \bmod 1, \tag{3}$$

where $B_2[P_j]$ denotes the P_j-th bit of B_2.

Step 3. Encrypting 16 consecutive pixels within $I^{(16)}(k)$. For the R, G, B values of each pixel, do the following operations to get the corresponding cipher-values R^*, G^* and B^*.

First, iterate the second Logistic map to extract K_{10} chaotic states $\{\hat{Y}_j\}_{j=1}^{K_{10}}$, by excluding chaotic states falling out of the interval $[0.1, 0.9]$. Then, encrypt the R, G, B values simultaneously according to the following equations:

$$R^* = E_1(R) = g_{K_4,K_5,K_7,K_8,\hat{Y}_{K_{10}}} \circ \cdots \circ g_{K_4,K_5,K_7,K_8,\hat{Y}_1}(R), \tag{4}$$

$$G^* = E_2(G) = g_{K_5,K_6,K_8,K_9,\hat{Y}_{K_{10}}} \circ \cdots \circ g_{K_5,K_6,K_8,K_9,\hat{Y}_1}(G), \tag{5}$$

$$B^* = E_3(B) = g_{K_6,K_4,K_9,K_7,\hat{Y}_{K_{10}}} \circ \cdots \circ g_{K_6,K_4,K_9,K_7,\hat{Y}_1}(B), \tag{6}$$

where $\circ$ denotes the composition of two functions and $g_{a_0,b_0,a_1,b_1,Y}(x)$ is a function under the control of Y as shown in Table 1.

In the sequel, $I^{*(16)}(k) = \{I^*(16k+0), \ldots, I^*(16k+i), \ldots, I^*(16k+15)\}$ will denote the corresponding blocks of cipher images.

Step 4. Updating sub-keys $K_1, \ldots, K_9$. Do the following updating operation for $i = 1 \sim 9$:

$$K_i = (K_i + K_{10}) \bmod 256, \tag{7}$$

[1] In Sec. 2 of [18], the interval is $[0.1, 0.9]$ and $P_j = \lfloor 23(\hat{X}_j - 0.1)/0.8 \rfloor + 1$. However, following this process, $P_j = 24$ when and only when $\hat{X}_j = 0.9$, which becomes a rare event and conflicts with the requirement that P_i has a roughly uniform distribution over $\{1, \ldots, 24\}$. Therefore, in this paper we changed the original process in [18] to a more reasonable one. Note that such a change does not influence the performance of the encryption scheme.

Table 1
The definition of $g_{a_0,b_0,a_1,b_1,Y}(x)$, where $\overline{x}$ denotes the bitwise complement of x, and $\oplus$ denotes the bitwise XOR operation.

$Y \in$	$g_{a_0,b_0,a_1,b_1,Y}(x)=$	$g^{-1}_{a_0,b_0,a_1,b_1,Y}(x)=$
$[0.10, 0.13) \cup [0.34, 0.37) \cup [0.58, 0.62)$	$\overline{x} = x \oplus 255$	
$[0.13, 0.16) \cup [0.37, 0.40) \cup [0.62, 0.66)$	$x \oplus a_0$	
$[0.16, 0.19) \cup [0.40, 0.43) \cup [0.66, 0.70)$	$(x + a_0 + b_0) \bmod 256$	$(x - a_0 - b_0) \bmod 256$
$[0.19, 0.22) \cup [0.43, 0.46) \cup [0.70, 0.74)$	$\overline{x \oplus a_0} = x \oplus \overline{a_0}$	
$[0.22, 0.25) \cup [0.46, 0.49) \cup [0.74, 0.78)$	$x \oplus a_1$	
$[0.25, 0.28) \cup [0.49, 0.52) \cup [0.78, 0.82)$	$(x + a_1 + b_1) \bmod 256$	$(x - a_1 - b_1) \bmod 256$
$[0.28, 0.31) \cup [0.52, 0.55) \cup [0.82, 0.86)$	$\overline{x \oplus a_1} = x \oplus \overline{a_1}$	
$[0.31, 0.34) \cup [0.55, 0.58) \cup [0.86, 0.90]$	$x = x \oplus 0$	

and then go to Step 2 and encrypt the next block until the whole plain-image is exhausted.

The decryption procedure is similar to the above encryption procedure, except that Eqs. (4)∼(6) in Step 3 are replaced by the following ones:

$$R = E_1^{-1}(R^*) = g^{-1}_{K_4,K_5,K_7,K_8,\hat{Y}_1} \circ \cdots \circ g^{-1}_{K_4,K_5,K_7,K_8,\hat{Y}_{K_{10}}}(R^*), \tag{8}$$

$$G = E_2^{-1}(G^*) = g^{-1}_{K_5,K_6,K_8,K_9,\hat{Y}_1} \circ \cdots \circ g^{-1}_{K_5,K_6,K_8,K_9,\hat{Y}_{K_{10}}}(G^*), \tag{9}$$

$$B = E_3^{-1}(B^*) = g^{-1}_{K_6,K_4,K_9,K_7,\hat{Y}_1} \circ \cdots \circ g^{-1}_{K_6,K_4,K_9,K_7,\hat{Y}_{K_{10}}}(B^*), \tag{10}$$

where $g^{-1}_{a_0,b_0,a_1,b_1,Y}(x)$ is the inverse function of $g_{a_0,b_0,a_1,b_1,Y}(x)$ with respect to x as shown in Table 1.

3 Cryptanalysis

3.1 Two remarks about Pareek et al.'s scheme

To facilitate the description of the discussion afterwards, we first analyze two properties of the scheme under study in this subsection. One is about the subkey update, and the other is about the essential equivalent presentation form of the encryption function.

To improve the security of the scheme, the original authors introduce an update mechanism for sub-keys, shown in Eq. (7). Obviously, the sequence of the updated sub-keys produced with such a mechanism is periodic. So, assuming that the period is T, then the $MN/16$ plain pixel-blocks $\{I^{(16)}(k)\}_{k=0}^{MN/16-1}$ can be divided into T groups $\left\{ \bigcup_{k=0}^{N_T-1} I^{(16)}(T \cdot k + j) \right\}_{j=0}^{T-1}$, where $N_T = MN/(16T)$.

For blocks in the same group, the update mechanism of sub-keys is disabled, namely the $\frac{1}{T}$ of the whole plain-image is encrypted with fixed sub-keys.

With respect to the encryption function, observing Table 1, one can see that each encryption sub-function can be represented in the following two formats:

(1) $g_{a_0,b_0,a_1,b_1,Y}(x) = x \oplus \alpha$, where $\alpha \in \{0, 255, a_0, a_1, \overline{a_0}, \overline{a_1}\}$;
(2) $g_{a_0,b_0,a_1,b_1,Y}(x) = x \dotplus \beta$, where $\beta \in \{a_0 \dotplus b_0, a_1 \dotplus b_1\}$, and $x \dotplus c$ denotes $(x + c) \bmod 256$ (the same hereinafter).

Since both operations verify $(x \oplus \alpha_1) \oplus \alpha_2 = x \oplus (\alpha_1 \oplus \alpha_2)$ and $(x \dotplus \beta_1) \dotplus \beta_2 = x \dotplus (\beta_1 \dotplus \beta_2)$, consecutive sub-encryption-functions of the same type can be combined together. As a result, each encryption function $E_i(x)$ is a composition of len sub-functions $\{G_j(x)\}_{j=1}^{len}$ with $len \leq K_{10}$, where $G_j(x) = x \oplus \alpha_{\lfloor j/2 \rfloor + 1}$ or $x \dotplus \beta_{\lfloor j/2 \rfloor + 1}$. According to the type of $G_1(x)$, $E_i(x)$ has two different formats:

- $E_i(x) = (\cdots(((x \oplus \alpha_1) \dotplus \beta_1) \oplus \alpha_2) \dotplus \beta_2) \oplus \cdots$;
- $E_i(x) = (\cdots(((x \dotplus \beta_1) \oplus \alpha_1) \dotplus \beta_2) \oplus \alpha_2) \dotplus \cdots$.

Since $G_j(x)$ is a multiple composition of functions $g_{a_0,b_0,a_1,b_1,Y}(x)$ of the same kind, one can easily deduce that

$$\alpha_i \in \mathbb{A} = \{0, 255, a_0, a_1, \overline{a_0}, \overline{a_1}, a_0 \oplus a_1, \overline{a_0} \oplus a_1\}$$

and

$$\beta_i \in \mathbb{B} = \{z_1(a_0 \dotplus b_0) \dotplus z_2(a_1 \dotplus b_1) | z_1, z_2 \in \{0, \ldots, K_{10}\}\}.$$

3.2 Analysis of the key space

In this subsection, we report some *invalid keys*, *weak keys* and *partially equivalent keys* existing in the encryption scheme under study. The term *invalid key* denotes a key that cannot ensure the successful working of the encryption scheme. A *weak key* is a key that displays a security hole. The term *partially equivalent keys* denotes those keys that work as the same key for some part of the plain-image. When estimating the key space, invalid keys and weak keys should be excluded; equivalent keys should be counted as one single key.

3.2.1 Invalid keys for $K_1 \sim K_9$

When $X_0 = 0$ or $Y_0 = 0$, the Logistic maps will fall into the fixed point 0, which disables the encryption process due to the lack of chaotic states lying in $[0.1, 0.9]$.

Observing Eq. (2), one can easily see that $X_0 = 0$ under the following subkeys: $K_4 = K_5 = K_6 = K_7 = K_8 = K_9 = 0$; $K_4 = K_5 = K_6 = 255, K_7 = K_8 = K_9 = 0$; $K_4 = K_5 = K_6 = 0, K_7 = K_8 = K_9 = 255$; $K_4 = K_5 = K_6 = K_7 = K_8 = K_9 = 255$. In addition, there is a set of combinations of K_4, K_5 and K_6 making

$$X_0 = \left(\frac{\sum_{i=4}^{6} K_i \cdot 2^{8(i-4)} + K_s/3 \cdot 2^{19}}{2^{24}} \right) \bmod 1 = \left(\frac{2^{24}}{2^{24}} \right) \bmod 1 = 0, \tag{11}$$

when $3|K_s$ and $K_s > 0$, where $K_s = \sum_{i=7}^{9}((K_i \bmod 16) + \lfloor K_i/16 \rfloor)$. For example, $K_4 = 0, K_5 = 0, K_6 = 248, K_7 = 3, K_8 = K_9 = 0$ verifies the previous equation. We have counted the cases satisfying this equation and obtained 41907 possibilities. [2]

For the k-th plain-block $I^{(16)}(k)$, the initial value Y_0 will be equal to zero definitely when the current values of K_1, K_2, K_3 are as follows: $K_1 = K_2 = K_3 = 0$; $K_1 = K_2 = K_3 = 255$. In addition, for any B_2, $Y_0 = 0$ when $(B_2 + \sum_{j=1}^{24} B_2[P_j] \cdot 2^{j-1}) = 2^{24}$. Assuming the distribution of P_j is uniform, the probability of this event can be counted as

$$p_s = (\frac{m}{24})^n \cdot (\frac{24-m}{24})^{24-n}, \tag{12}$$

where m and n are the numbers of zero bits in binary presentation of B_2 and $\overline{B_2} + 1$ respectively. Note that n and m depend on B_2 but they may be considered independent since there is no way to compute n from m. For example, when $K_1 = 0, K_2 = 0, K_3 = 128$ and $B_2 = 8388608$, $p_s = (\frac{23}{24})^{23} \cdot (\frac{24-23}{24})^{24-23} \approx 0.0157$. Although ordinarily the value of p_s is extremely small (See Fig. 1), the potential hole exists for the encryption of any plain block with any current secret key, i.e., any secret key may act as an invalid one [3].

3.2.2 Weak keys for K_{10}

In the scheme under study, the update of sub-keys $K_1 \sim K_9$ and the iteration number of sub-functions $g_{a_0,b_0,a_1,b_1,Y}(x)$ are both controlled by the sub-key K_{10}. In the following, we discuss the weak key problems relative to K_{10} and related to this two tasks.

Observing Eq. (7) one may see that the update of subkeys $K_1 \sim K_9$ has an inherent weakness derived from the following well known fact:

[2] Since there is no simple expression, we resort to enumerating all possible cases via a computer program.

[3] In the discussion afterwards and experimental implementation, we adopt a tiny fluctuation, set Y_0 as $\frac{1}{2^{24}}$ when $Y_0 < \frac{1}{2^{24}}$, to avoid this problem.

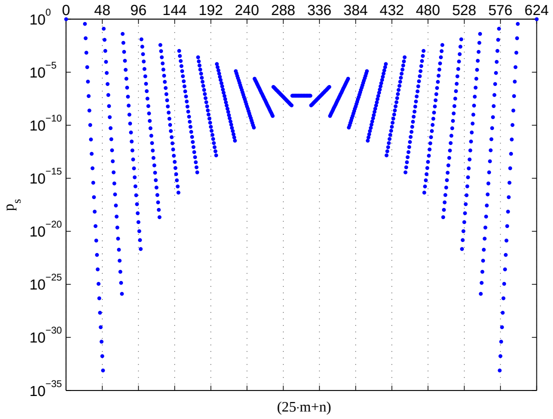


Fig. 1. $(25 \cdot m + n)$ vs. p_s. Note that there is a deterministic relation between (m, n) and $(25 \cdot m + n)$ since the scopes of m and n are both $[0, 24]$.

Fact 1 *for $x, a \in \{0, \ldots, 255\}$, the integer sequence $\{y(i) = (x + ai) \bmod 256\}_{i=0}^{\infty}$, has period $T = 256/\gcd(a, 256)$.*

Thus, the possible values for the period of the sequence of updated sub-keys is 2^i, with $i = 1 \sim 8$. For some values of K_{10}, the period can be very small, which weaken the updating mechanism considerably. The situation is specially dramatic for $K_{10} = 128$, where the period is two.

Given K_{10}, and following the notation introduced in Sec. 3.1, the blocks of the group $\left\{\bigcup_{k=0}^{N_T-1} I^{(16)}(T \cdot k + j)\right\}$ for any $j \in \{0, \ldots, T-1\}$ will compose a strip of width 16 when $16T | N$, which is a sub-image encrypted with fixed sub-keys.

Now, let us study the weak key problem of K_{10} for the task of controlling the iteration number of sub-functions $g_{a_0,b_0,a_1,b_1,Y}(x)$.

When K_{10} is too small, the probability for a pixel to remain unchanged is not negligible. This situation may yield a leak of visual information in a channel or in the whole plain-image. The worst case occurs when $K_{10} = 1$, where, on the average, one eighth of the image is not encrypted. If moreover the values of a_0, a_1, b_0, b_1 make some sub-encryption-function collapse to x, this probability will increase. Intuitively, sufficiently large K_{10} may guarantee that the encryption functions $E_i(x)$ are not trivial.

To illustrate this problem, we disabled the secret key update mechanism, and calculated the number of different values in the encryption result of a 512×512 image with fixed value zero. We found that the possible number may be smaller than or equal to 128 when $(a_0 \dot{+} b_0)$ and $(a_1 \dot{+} b_1)$ are even at the same time. Among 50,000 times random experiments, where sub-keys $K_1 \sim K_9$ were selected randomly and $K_{10} = 255$ there were 1,143 such cases (account for

2.286%). For the secret sub-keys $K_1 \sim K_9$ that made the number of possible encryption results reach 256 under the sufficient large K_{10}, the relation between the mean value of the possible numbers and K_{10} is shown in Fig. 2.

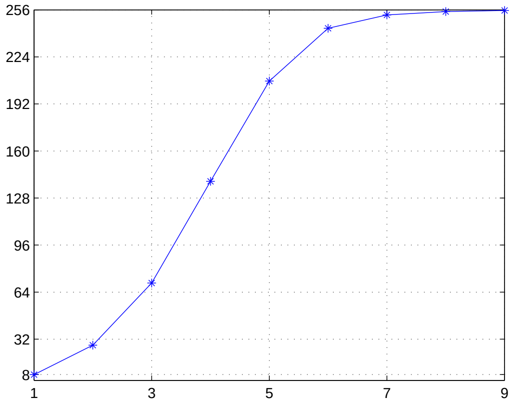


Fig. 2. The value of K_{10} vs. the mean value of the numbers of possible encryption result.

From the previous discussion we conclude two criteria for a cryptographically strong sub-key K_{10}: 1) K_{10} is an odd integer; 2) K_{10} is sufficiently large (at least larger than 8).

3.2.3 Weak keys for $K_4 \sim K_9$

Observing Table 1, one can see that the encryption sub-functions verify $g_{a_0,a_1,b_0,b_1,y}$ $(x) = x$ or $g_{a_0,a_1,b_0,b_1,y}(x) = \bar{x}$ when the following requirements are satisfied:

$$a_0, a_1 \in \{0, 255\} \text{ and } a_0 + b_0 \equiv a_1 + b_1 \equiv 0 \pmod{256}. \tag{13}$$

In this case, the three composition encryption functions, $E_1(x)$, $E_2(x)$ and $E_3(x)$, are also be x or the identity of the bitwise complement. Assuming that the chaotic trajectory of the second Logistic map has an uniform distribution in the interval $[0.1, 0.9]$, one can check that the probability of $g_{a_0,a_1,b_0,b_1,y}(x) = \bar{x}$ is $p = 3/8$. Then, according to Lemma 1 (note that $\bar{x} = x \oplus 255$), $\forall i = 1 \sim 3$, the probabilities of $E_i(x) = \bar{x}$ and $E_i(x) = x$ are $(1 - (1/4)^n)/2$ and $(1 + (1/4)^n)/2$ respectively, where $n = Y_{10}$. This means that, for n sufficiently large, about half of all plain-pixels from the group of blocks $\{\bigcup_{k=0}^{N_T-1} I^{(16)}(T \cdot k + 0)\}$ are not encrypted at all, which may reveal some visual information in the plain-image. In the experiment shown in Fig. 3, 49.9% of the pixels of the vertical strips are failed to be encrypted.

Lemma 1 *Given $n > 1$ functions, $f_1(x), \ldots, f_n(x)$, assume that each function is $x \oplus a$ with probability p and is x with probability $1 - p$, where $a \in \mathbb{Z}$. Then,*

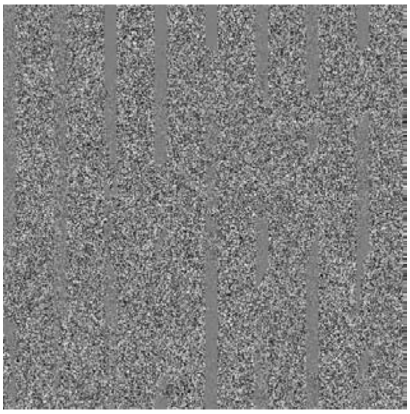


Fig. 3. The encryption result when $K =$ "$3C1DE8FF0151FF012840$" (represented in hexadecimal format, the same hereinafter).

the probability of the composition function $F(x) = f_1 \circ \cdots \circ f_n(x) = x \oplus a$ *is* $P = (1 - (1 - 2p)^n)/2$*. If* $p < 1/2$ *then* $P < 1/2$*.*

Proof: Assuming that $k = \lceil n/2 \rceil$, then $n = 2k$ if it is an even integer and $n = 2k - 1$ when it is an odd integer. To ensure $F(x) = f_1 \circ \cdots \circ f_n(x) = x \oplus a$, the number of sub-functions that are equal to $x \oplus a$ should be an odd integer. So, we have

$$\begin{aligned} P &= \sum_{i=1}^{k} \binom{n}{2i-1} p^{2i-1}(1-p)^{n-(2i-1)} \\ &= (1-p)^n \cdot \sum_{i=1}^{k} \binom{n}{2i-1} (p/(1-p))^{2i-1} \\ &= (1-p)^n \cdot \frac{(1+p/(1-p))^n - (1-p/(1-p))^n}{2} \\ &= (1-(1-2p)^n)/2 < 1/2. \end{aligned}$$

This completes the proof of the lemma. ■

By letting Eq. (13) hold for function $E_i(x)$, $i = 1 \sim 3$, we can get a list of weak keys as shown in Table 2.

Actually, the above analysis can be further generalized to obtain families of not so weak keys satisfying one of the following conditions, which are conservatively estimated according to some random tests.

1. $\#(\mathbb{A}) < 7$, where $\#(\mathbb{S})$ denote the cardinality of set $\mathbb{S}$ (The same hereinafter);
2. More than half elements of $\mathbb{A}$ are too small, such as $a_0, a_1, a_0 \oplus a_1$ are less than 10 at the same time;
3. $\#(\mathbb{B}) < 256$.

Table 2
Some weak keys that cause leaking of visual information.

Weak keys	Visual information leaked from
$K_4 = K_5 = K_7 = K_8 = 0$	Channel R
$K_4 = K_7 = 255,\ K_5 = K_8 = 1$	
$K_5 = K_6 = K_8 = K_9 = 0$	Channel G
$K_5 = K_8 = 255,\ K_6 = K_9 = 1$	
$K_4 = K_6 = K_7 = K_9 = 0$	Channel B
$K_6 = K_9 = 255,\ K_4 = K_7 = 1$	
$K_4 = K_5 = K_6 = K_7 = K_8 = K_9 = 0$	the whole plain-image

To display this potential problem efficiently, we deliberately discard the update function Eq. (7), and show one example in Fig. 4.

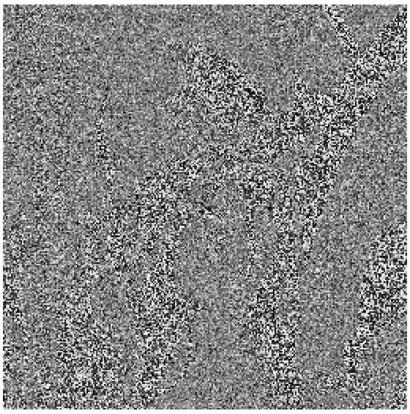


Fig. 4. The red channel of encryption result when Eq. (7) is disabled and $K =$ "$3C1DE8FF0A51FF0A2840$".

From the most conservative point of view, a cryptographically strong key should ensure that all the sub-encryption-functions shown in Table 1 are not weakened. This means that a strong key should satisfy the following criteria: 1) $\#(\mathbb{A}) = 7$, 2) the value of elements in $\mathbb{A}$ is larger than 10; 3) $\#(\mathbb{B}) = 256$.

3.2.4 Partially equivalent keys for $K_7 \sim K_9$: Class 1

Observing Eq. (2), one can see that the value of X_0 remains unchanged if the following segments of K_7, K_8, K_9 exchange their values: $K_7 \bmod 16$, $\lfloor K_7/16 \rfloor$, $K_8 \bmod 16$, $\lfloor K_8/16 \rfloor$, $K_9 \bmod 16$, $\lfloor K_9/16 \rfloor$. Now let us investigate what will happen if we exchange $K_9 \bmod 16$ and $\lfloor K_9/16 \rfloor$, i.e., exchange the upper half and the lower half of K_9. In this case, since the encryption of the red value of each pixel is independent of K_9, the red channel of the cipher-image will remain unchanged. For the plain-image shown in Fig. 5, this phenomenon is shown in Fig. 6. Similar results also exist for K_7 and K_8, which correspond to unchanged blue and green channels of the plain-image, respectively.

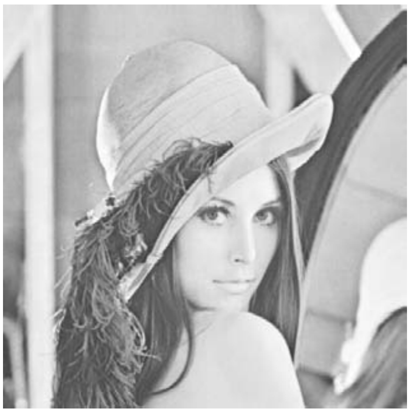


Fig. 5. The red channel of plain-image “Lenna”(displayed as a gray-scale image).

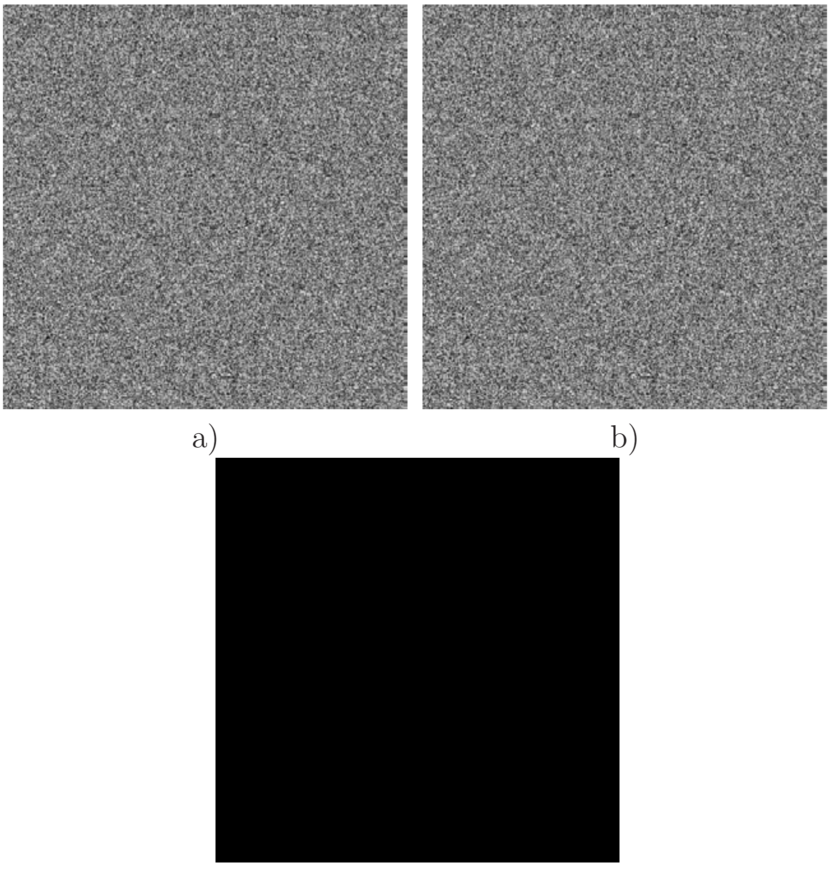


Fig. 6. The encryption results of two partially equivalent keys of Class 1, $K =$ “$1A84BCF35D70664E\underline{47}50$” and $\widetilde{K} =$ “$1A84BCF35D70664E\underline{74}50$”: a) the red channel of the cipher-image corresponding to K; b) the red channel of the cipher-image corresponding to $\widetilde{K}$; c) the difference image.

3.2.5 Partially equivalent keys for $K_7 \sim K_9$: Class 2

As remarked in Sec. 3.1, one can see that each sub-encryption-function $g_{a_0,a_1,b_0,b_1,Y}(x)$ can be represented as one of the following two kinds of functions: $x \oplus \alpha$, and $x \dot{+} \beta$. The combination of this fact and the following two properties of octets will lead us to construct another class of partially equivalent keys. The following property is immediate:

Fact 2 *$\forall\ a \in \{0, \ldots, 255\}$, $a \oplus 128 = a \dot{+} 128$.*

The previous property and the associativity of modular arithmetic yields to the following:

Fact 3 *$\forall\ a, b \in \mathbb{Z}$, the following result is true: $(a \oplus 128) \dot{+} b = (a \dot{+} b) \oplus 128$.*

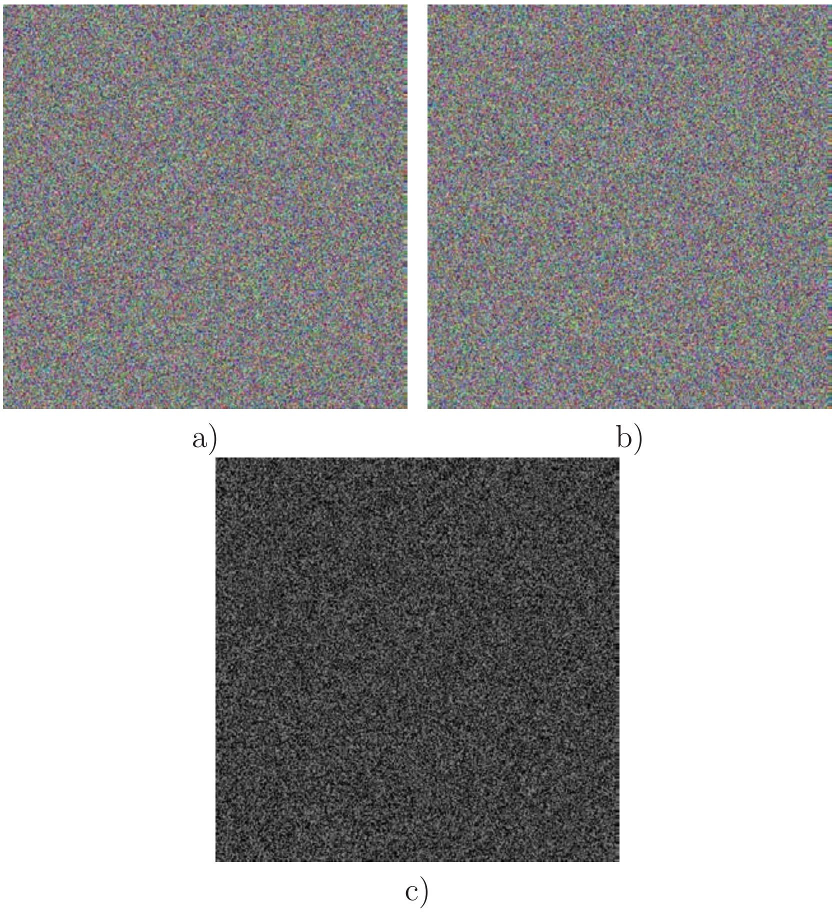
Fact 3 means that a change in the MSB (most significant bit) of x, a_0, a_1, b_0, b_1 of any sub-encryption-function $g_{a_0,a_1,b_0,b_1,Y}(x)$ is equivalent to XORing 128 on the output of the composition function $E_i(x)$.

Next, let us investigate how to use this fact to figure out another class of partially equivalent keys about $K_7 \sim K_9$. Choose any two sub-keys from $K_7 \sim K_9$. For instance take K_7 and K_8. Then, given a secret key K that satisfies $K_7 < 128$ and $K_8 \geq 128$ (or, $K_7 \geq 128$ and $K_8 < 128$), let us change it into another key $\widetilde{K}$ by setting $\widetilde{K_7} = K_7 \oplus 128$ and $\widetilde{K_8} = K_8 \oplus 128$. From Eq. (2), it is easy to see that X_0 remains the same for the two keys. This means that the two Logistic maps have the same dynamics throughout the encryption procedure for the two keys, and the difference on ciphertexts is only determined by the MSB-changes of K_7 and K_8. In the following, we consider the three color channels separately.

First, let us consider the encryption of the green channel of the plain-image, in which K_7 is not involved at all. Assuming that the chaotic trajectory $\{Y_i\}$ is distributed uniformly within the interval $[0.1, 0.9]$, one can see that the probability that K_8 has an effect on each encryption sub-function is $p = 3/8$. If K_8 appears an even number of times of the K_{10} total encryption sub-functions, then the value of $E_2(G)$ will remain the same for the two keys; otherwise $E_2(G)$ changes its MSB. Thus, using the same deduction as the given in the proof of Lemma 1, we can get immediately the probability of $E_2(G)$ to remain unchanged as $P_2 = (1 + (1 - 2p)^{K_{10}})/2 = (1 + (1 - 3/4)^{K_{10}})/2$. This means that more than half of all green pixel values in the ciphertexts are identical for the two keys K and $\widetilde{K}$ in probability.

For the blue channel, the condition is completely similar, and one can get the probability of $E_3(B)$ to remain unchanged as $P_3 = (1 + (1 - 3/4)^{K_{10}})/2 = P_2$. For the red channel, both K_7 and K_8 are involved, but their differences are neutralized for the sub-encryption-function $(x + K_7 + K_8) \bmod 256$. So, the probability that the differences in K_7 and K_8 have an effect is reduced to be $p = 2/8 = 1/4$. Then, one can get the probability of $E_1(R)$ to remain unchanged as $P_1 = (1 + (1 - 1/2)^{K_{10}})/2 > P_2 = P_3$.

To verify the above theoretical results, we made some experiments for a plain-image of size 512×512 and some of the results are shown in Figure 7. The numbers of the same elements in red, green and blue channel of Figs. 7a) and b) are 131241, 130864 and 131383 respectively. The XOR difference between red channel of Figs. 7a) and b) is shown in Fig. 7c) as an example, which is a $\{0, 128\}$ binary image.



a) b)

c)

Fig. 7. The encryption results of two partially equivalent keys of Class 2, K = "$1A93DF25CF78DC44E160$" and $\widetilde{K}$ = "$1A93DF25CF78\underline{5}C\underline{C}4E160$": a) The cipher-image corresponding to K; b) The cipher-image corresponding to $\widetilde{K}$; c) the red channel of the XOR difference image.

3.2.6 Reduction of the key space

Based on the above analysis, we briefly summarize here the influence of invalid, weak and equivalent keys in the key space. The result is shown in Table 3. From the table, we can roughly estimate that the size of key space is reduced to 2^{75}, which is little smaller than 2^{80}, the one claimed in [18, Sec. 3.3].

Table 3
Analysis of key space.

Sub-keys	Reduced numbers	Reason
$K_1 \sim K_3$	2	$Y_0 = 0$
$K_4 \sim K_9$	41911	$X_0 = 0$
$K_4 \sim K_9$	$2 \cdot 254^3 + 2^6$	$\#(\mathbb{A}) < 7$
$K_4 \sim K_9$	$2 \cdot 10^6$	Four elements of $\mathbb{A}$ are less than 10
$K_4 \sim K_9$	$(\sum_{i=1}^{128}(2i))^6$	$\#(\mathbb{B}) < 256$
$K_7 \sim K_9$	$(256^3 - \frac{256-16}{2})^3$	Equivalent key of class 1
$K_7 \sim K_9$	$3 \cdot 128^3$	Equivalent key of class 2
K_{10}	128	$T < 256$

3.3 Vulnerabilities of the cryptosystem

In this subsection, we discuss some security problems of the encryption scheme under study.

3.3.1 Potential security holes for the encryption function $E_i(x)$

As remarked in Sec. 3.1, the combination of some encryption sub-functions may produce some simplifications in the resultant encryption function. If the encryption sub-function is of the type $x \oplus \alpha$, the iteration of an even number of times of the same function will become the identity. More generally, the result of any iteration of functions of the kind $x \oplus \alpha$ with $\alpha \in \mathbb{A}$, is still a function $x \oplus \alpha$ with $\alpha \in \mathbb{A}$. If the encryption sub-function is of the type $x \dotplus \beta$, the T and $\frac{T}{2}$ times consecutive iteration of it will become to $x \dotplus 0$ and $x \dotplus 128$, respectively.

The direct consequence of these holes is that the number of real effective operations in $E_i(x)$, *len*, may be much less than K_{10}. Assuming that the distribution of chaotic states generated by iterating Logistic map is uniform, we can calculate

$$Prob(len = K_{10}) = \begin{cases} 2 \cdot (\frac{5}{8} \cdot \frac{1}{4})^{\frac{K_{10}}{2}} & \text{when } K_{10} \text{ is even,} \\ (\frac{5}{8} \cdot \frac{1}{4})^{\lfloor \frac{K_{10}}{2} \rfloor}(\frac{5}{8} + \frac{1}{4}) & \text{when } K_{10} \text{ is odd.} \end{cases} \quad (14)$$

From the above equation, we can see that the probability is extremely small when K_{10} is large enough. The estimation of the probability for len to be a given value is very complex. Alternatively, we carry out a number of random experiments for a 512×512 plain image with fixed K_{10}. Figure 8 shows the bounds of distribution of len for 100 times random experiments, in which $K_{10} = 66$, and other sub-keys are selected randomly.

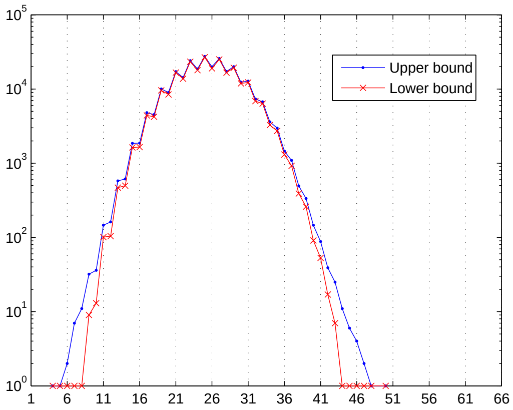


Fig. 8. The value of len vs. the number of sub-functions whose number of real operation is len.

From Fig. 8, it can be observed that len is much less than K_{10} in a relative large probability. These potential security holes mean that not every encryption step contribute to the security of the whole scheme, and even compromise it. It also help to explain why the slope of the line shown in Fig. 2 is so slow.

3.3.2 Non-uniformity of the cipher-image

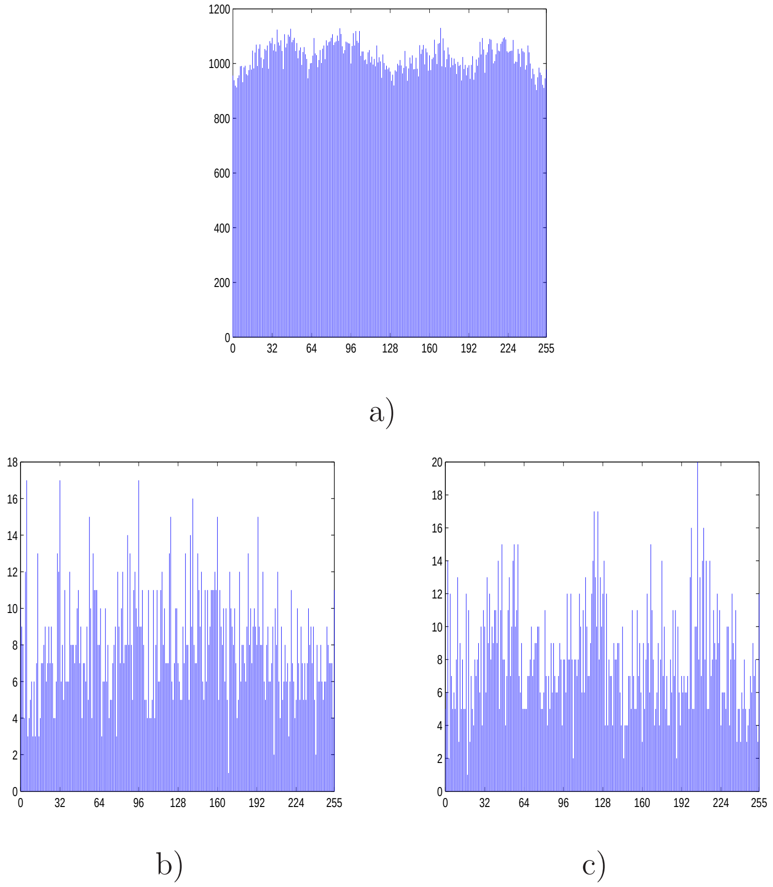
Although the original authors have validated the uniform histogram of the whole cipher-image in [18, Sec. 3.1], we found that the histograms of some sub-images of the cipher-image are not uniform.

As discussed in Sec. 3.1, the scheme under study works for T sub-images $\{\bigcup_{k=0}^{N_T-1} I^{(16)}(T \cdot k + j)\}_{j=0}^{T-1}$ with fixed secret key. Furthermore, from the above sub-subsection, we can guess that the distribution of pixels in $\bigcup_{k=0}^{N_T-1}\{I^{*(16)}(T \cdot k+j)\}$ is not uniform enough. To validate this estimation, we made a number of random experiments. The secret key "ABCDEF0123456789FF05", the one also used in [18, Fig. 1], was used to encrypt a 512×512 plain-image "Lenna." The histogram of the red channel of cipher-image is shown in Fig. 9 a). As the discussion in Sec. 3.2.5, $\forall\, x \in \left\{I^{*(16)}(T \cdot k + j) \oplus I^{*(16)}(T \cdot k + j + T/2)\right\}$,

$x \in \{0, 128\}$. So we divide the cipher-image into $T/2$ parts,

$$\left\{\bigcup_{k=0}^{N_T-1}\left\{I^{*(16)}(T\cdot k+j)\cup I^{*(16)}(T\cdot k+j+T/2)\right\}\right\}_{j=0}^{T/2-1},$$

and show the distributions of the 0-th and 21-th one in Figs. 9b) and c) respectively. Just as expected, the distribution of the whole cipher-image is relative uniform, however the distributions of the two sub-images are not uniform, which make it feasible for cipher-text only attack. Note that this potential flaw exists for any value of K_{10} since a larger value of T means more distribution diagrams with weaker detail, and a smaller value of T means less distribution diagrams with stronger detail.



a)

b)

c)

Fig. 9. Distribution of the cipher-image: a) the histogram of the whole cipher-image; b) the histogram of the 0-th part of Fig. 9a); c) the histogram of the 21-th part of Fig. 9a).

3.3.3 Low sensitivity to plain-image

Unfortunately, the scheme under study fails to satisfy the property on the MSB of each pixel value since $E_i(x \oplus 128) = E_i(x) \oplus 128$ from Fact 3. So, the change of the MSB of any pixel value only lead to the change of the one of corresponding cipher-pixel value, which demonstrate the extremely weak sensitivity.

3.4 Guessing K_{10} with a chosen plain-image

As remarked in Sec. 3.1, all 16-pixel blocks in $\left\{I^{(16)}(T\cdot k+j)\right\}_{k=0}^{N_T-1}$ are encrypted with the same sub-keys. If these blocks also correspond to the same values of Y_0, then all the three encryption functions for R, G, B channels will become identical. Precisely, given two identical blocks, $I^{(16)}(k_0)$ and $I^{(16)}(k_1)$, one can see that the corresponding cipher-blocks will also be identical, in the case that the following two requirements are satisfied:

(A) the distance of the two blocks is a multiple of T, i.e., $(k_0-k_1)\mid T$;
(B) $Y_0^{(k_0)}=Y_0^{(k_1)}$, where $Y_0^{(k_0)}$ and $Y_0^{(k_1)}$ denote the value of Y_0 corresponding to the two 16-pixel blocks.

If the probability of the two cipher-blocks to be identical is sufficiently large, we may use the distance between them to determine the value of T and narrow the search space of K_{10}. However, the two cipher-blocks may also be identical by accident not satisfying one of the above requirements, but we will show later that the probability of such a false event is much smaller than the probability of the real event so that it can be simply neglected in practice.

Next, let us calculate the probability of the above two requirements to hold simultaneously. From the definition of conditional probability, we have $Prob(A\cap B)=Prob(A)Prob(B|A)$. Under the assumption that the two blocks are chosen at random, one can easily deduce that $Prob(A)=1/T$. Since the values of $B_2=K_1K_2K_3$ are the same when $(k_0-k_1)\mid T$, from Eq. (3) one has

$$\begin{aligned}Y_0^{(k_0)}-Y_0^{(k_1)}&=\left(\frac{\sum_{k=1}^{24}\left(B_2\left[P_k^{(k_0)}\right]-B_2\left[P_k^{(k_1)}\right]\right)\times 2^{k-1}}{2^{24}}\right)\bmod 1,\\&=\sum_{k=1}^{24}\left(B_2\left[P_k^{(k_0)}\right]-B_2\left[P_k^{(k_1)}\right]\right)\times 2^{k-25},\end{aligned}$$

where $P_k^{(k_0)}$ and $P_k^{(k_1)}$ denote the values of P_k corresponding to the two blocks. Then, one can get $Prob(B|A)=Prob\left(B_2\left[P_k^{(k_0)}\right]=B_2\left[P_k^{(k_1)}\right],\forall k=1\sim 24\right)$. Assuming that each P_k has an uniform distribution over $\{1,\ldots,24\}$ and there are m 0-bits in the binary representation of B_2, one can deduce that the probability of $B_2\left[P_k^{(k_0)}\right]=B_2\left[P_k^{(k_1)}\right]$ is

$$p_0=\left(\frac{m}{24}\right)^2+\left(\frac{24-m}{24}\right)^2=1-\frac{m}{12}+\frac{m^2}{288}.$$

Further assuming that any two elements in $\{P_k\}_{k=1}^{24}$ are independent of each other, one has $Prob(B|A) = p_0^{24}$. Combining with the probability of B_2 having m 0-bits in its binary representation, one can determine the value of $Prob(B|A)$ when B_2 (i.e., the sub-key $K_1K_2K_3$) is generated at random:

$$\begin{aligned} Prob(B|A) &= \sum_{m=0}^{24} p_0^{24} \cdot \binom{24}{m} \Big/ 2^{24}, \\ &= \sum_{m=0}^{24} \left(1 - \frac{m}{12} + \frac{m^2}{288}\right)^{24} \cdot \binom{24}{m} \Big/ 2^{24} \approx 2^{-18.3}. \end{aligned}$$

Then, one can finally get $Prob(A \cap B) = Prob(A) \cdot Prob(B|A) \approx 2^{-18.3}/T$.

Note that the above theoretical analysis is based on the following idealized assumption: $P_1 \sim P_{24}$ are independent and identically-distributed random variables with an uniform distribution over $\{1, \ldots, 24\}$. However, in reality, this assumption is generally not true, and as a result the actual value of $Prob(A \cap B)$ may be much larger than the theoretically estimated value. We have made a number of experiments to study the non-uniform distribution of the elements of the sequence $\{P_k\}_{k=1}^{24}$, i.e., an approximate distribution of trajectories of the Logistic map in the interval $[0.1, 0.9]$. All the distributions are similar to each other, so only the trajectory generated by $Y_0 = 0.35$ is shown in Fig. 10 for illustration. Obviously, the non-uniform distribution of the chaotic trajectory will improve the correlation between elements in $\{P_k\}_{k=1}^{24}$.

To validate this point, we carried out 100 experiments computing $\{P_k^{(i)}\}_{i=1}^{16384}$ (the blocks for encryption of a 512×512 plain-image) with random generated keys. In $\{P_k^{(i)}\}_{i=1}^{16384}$, we consider all possible pairs of sequences $\{P_k\}_{k=1}^{24}$ (a number of $\binom{16384}{2} = 134209536 \approx 2^{27}$ pairs) and count those pairs whose two sequences coincide in a given number k of elements. In Fig. 11 it is represented the result of this experiment. For any number k it is given the mean value of numbers of sequences coincident in the first k elements. As a comparison, the theoretical probability $p_k = \binom{24}{k}(\frac{1}{24})^k(\frac{23}{24})^{24-k}$ is also plotted in the same figure. Note that the number of the same elements between any pair of $\{P_k\}_{k=1}^{24}$ is less than or equal to 22 in the experiment.

To reveal the actual value of $Prob(A \cap B)$, we carry out 50,000 random tests with a plain-image of size 1024×768, where $K_{10} = 162$ and $K_1 \sim K_9$ are selected randomly. We count the mean value and standard deviation of the results and obtain $2^{-24.87}$ and $2^{-21.42}$ respectively. The mean value is a little larger than the theoretical one $2^{-18.3}/T = 2^{-25.3}$, but the standard deviation means that the former is much larger than the latter for some cases.

Now let us analyze the probability of the false event, i.e., the probability that two cipher-blocks are identical when at least one of the two requirements is

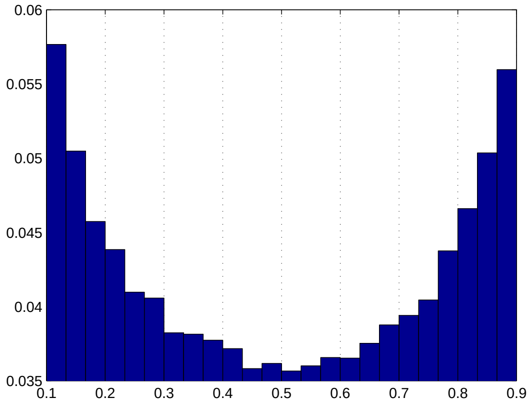


Fig. 10. Distribution of $\{P_k^{(i)}\}_{i=1}^{393216}$ when $Y_0 = 0.35$.

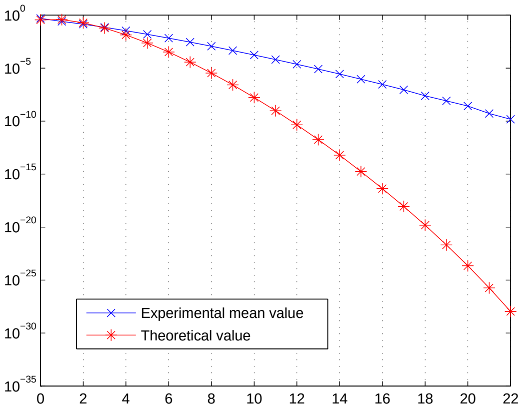


Fig. 11. The number of the same elements between blocks $\{P_k\}$ vs. corresponding probability.

not satisfied. Assuming that each pixel value has an uniform distribution over $\{0, \ldots, 255\}$ and that any two pixel values are statistically independent of each other, one can easily deduce that this probability is $(\frac{1}{256})^{16\times 3} = 2^{-384}$. As this probability is so tiny, we can completely neglect it in practice.

The fact that $Prob(A \cap B) \approx 2^{-24}$ means that an image of size 512 × 512 may be enough to carry out a chosen-plaintext attack. To validate the feasibility of the attack, we chose a special plain-image with fixed value, and check the corresponding cipher-image encrypted with a random secret key $K =$ "$2A84BCF35D70664E4740$". Then we found 9 pairs of identical blocks whose indices are listed in Table 4. Since all these indices should satisfy the requirement $(k_0 - k_1) \mid T$, we can get an upper bound of T by solving their greatest common divisor of the 9 index-differences. For the data shown in

Table 4, one can immediately get

$$\gcd(3161-1941, 7083-2015, 15255-3023, 9163-4159, 12113-5061,\\ 16355-5507, 12454-9166, 12259-9655, 13102-11090) = 4.$$

This immediately leads to $T \in \{2,4\}$. Then, from Fact 1, one has $\gcd(K_{10}, 256) \in \{128, 64\}$ and further gets $K_{10} \in \{64, 128, 192\}$. We can see that the size of the sub-key space corresponding to K_{10} is reduced to 3 from 256, which is a significant reduction.

Table 4
The indices of 9 pairs of identical blocks in the cipher-image corresponding to the plain-image of fixed value zero.

i	1941	2015	3023	4159	5061	5507	9166	9655	11090
j	3161	7083	15255	9163	12113	16355	12454	12259	13102

3.5 A Differential Chosen-Plaintext Attack

First, we prove some useful properties related to the composite functions $E_i(x)$. Such properties are the basis of the differential attack introduced in this subsection.

Theorem 1 *Let $F(x) = G_{2m+1} \circ \cdots \circ G_1(x)$ be a composite function defined over $\{0, \ldots, 255\}$, where $G_{2i}(x) = x \oplus \alpha_i$ for $i = 1 \sim m$, $G_{2i+1}(x) = x \dotplus \beta_i$ for $i = 0 \sim m$ and $\alpha_i, \beta_i \in \{0, \ldots, 255\}$. If $F(x) = x \oplus \gamma$ for some $\gamma \in \{0, \ldots, 255\}$, then $\gamma \in \{\oplus_{i=1}^m \alpha_i, (\oplus_{i=1}^m \alpha_i) \oplus 128\}$.*

Proof: First, let us introduce some notation. Let $x = \sum_{j=0}^{7} x_j \cdot 2^j$, $\alpha_i = \sum_{j=0}^{7} \alpha_{i,j} \cdot 2^j$, $\beta_i = \sum_{j=0}^{7} \beta_{i,j} \cdot 2^j$, and $F(x) = \sum_{j=0}^{7} F_j(x) \cdot 2^j$.

The proof is based on the following fact. If F verifies that $F(x) = x \oplus \gamma$ for some γ then, for any $i = 0 \sim 7$, the result of the computation of $F_i(x)$ depends only on the value of the i-th coordinate of x, that is, on x_i.

We are going to check the computation of $F(x)$ starting from the least significant bit. We write $F_0(x) = x_0 \dotplus \beta_{0,0} \oplus \alpha_{1,0} \dotplus \beta_{1,0} \oplus \cdots \oplus \alpha_{m,0} \dotplus \beta_{m,0}$. The calculation is carried out from the left to the right and the carry bits occurring for the operation $\dotplus$ will have an effect in other coordinates. Observe that, to compute the value of $F_0(x)$, one simply may write $F_0(x) = x_0 + \beta_{0,0} + \alpha_{1,0} + \beta_{1,0} + \cdots + \alpha_{m,0} + \beta_{m,0}$, where, in this context, the operation $+$ is the sum modulo 2 and it is equivalent to $\oplus$.

Let us study how the carry bits occurring at the first coordinate affect the second and other coordinates.

- If $\beta_{i,0} = 1$ for some i, then one and only one of the values of x_0 will generate a carry bit at i.
- When a carry bit occurs at $\beta_{i,0}$ for some i, then $\beta_{i,0} = 1$.

This implies that the cardinal of the set $\{i \mid \beta_{i,0} = 1\}$ is the sum of the carry bits occurring when $x_0 = 0$ and the carry bits occurring when $x_0 = 1$.

The effect of the carry bits on the second coordinate has to be the same when $x_0 = 0$ and when $x_0 = 1$. For that reason both numbers (the sum of the carry bits occurring when $x_0 = 0$ and when $x_0 = 1$) need to have the same parity and hence the cardinal of $\{i \mid \beta_{i,0} = 1\}$ is even. Then, $F_0(x) = x_0 \oplus \alpha_{1,0} \oplus \cdots \oplus \alpha_{m,0}$. This proves the theorem for the least significant bit.

This argument may be used for any bit but the most significant one. Without loss of generality, we may assume we are working with the sixth (least significant) bit. $F_5(x)$ does not depend only on $\{\alpha_{i,5}\}_{i=1}^{m}$ and $\{\beta_{i,5}\}_{i=0}^{m}$ but also on the ones corresponding to carry bits occurring in previous bits.

Suppose we evaluate the octet $x = (x_7, x_6, x_5, 0, 0, 0, 0, 0)$. The evaluation of the first five bits will produce carry bits on the sixth position. Hence, $F_5(x) = x_5 \dotplus \bar{\beta}_{0,5} \oplus \alpha_{1,5} \dotplus \bar{\beta}_{1,5} \oplus \cdots \oplus \alpha_{m,5} \dotplus \bar{\beta}_{m,5}$ where $\bar{\beta}_{i,5}$ and $\beta_{i,5}$ disagree only when the effect of a carry bit produced in a previous bit is visible at the sixth bit. Observe that the final value of $F_5(x)$ does not depend on any x_j with $j \neq 5$.

Repeating the argument for the least significant bit, and working only with octets with the first five bits equal to zero, we conclude that the set $\{i \mid \bar{\beta}_{i,5} = 1\}$ has even cardinal.

Now take any other octet $x = (x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ and consider $F_5(x) = x_5 \dotplus \breve{\beta}_{0,5} \oplus \alpha_{1,5} \dotplus \breve{\beta}_{1,5} \oplus \cdots \oplus \alpha_{m,5} \dotplus \breve{\beta}_{m,5}$, where $\breve{\beta}_{i,5}$ and $\beta_{i,5}$ disagree only when the effect of a carry bit produced in a previous bit is visible at the sixth bit. Notice that $(x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ is fixed and the definition of $\breve{\beta}_{i,5}$ depends on $(x_4, x_3, x_2, x_1, x_0)$. Since this sum only depends on x_5 we conclude that the parity of the sets $\{i \mid \bar{\beta}_{i,5} = 1\}$ and $\{i \mid \breve{\beta}_{i,5} = 1\}$ have to coincide and hence is even. This implies that $F_5(x) = x_5 \oplus \alpha_{1,5} \oplus \cdots \oplus \alpha_{m,5}$ for any x.

This argument does not work for the most significant bit because the carry bits are lost and do not have any effect about the result of the function.

Note that the above analysis is independent of the value of x. So $F(x) = x \oplus \gamma$, where $\gamma \in \{\oplus_{i=1}^{m} \alpha_i, (\oplus_{i=1}^{m} \alpha_i) \oplus 128\}$, which completes the prove of this theorem. ■

Corollary 1 *If there exists* $\gamma \in \{0, \ldots, 255\}$ *such that* $E_i(x) = x \oplus \gamma$, *then* $\gamma \in \left\{\bigoplus_{i=1}^{\lfloor K_{10}/2 \rfloor} \alpha_i, \left(\bigoplus_{i=1}^{\lfloor K_{10}/2 \rfloor} \alpha_i\right) \oplus 128\right\}$.

Proof: Let us consider the following three different cases to prove the corollary.

(1) When $G_1(x) = x \dot{+} \beta_1$ and $G_{K_{10}}(x) = x \dot{+} \beta_{\lfloor K_{10}/2 \rfloor}$, this corollary is the same as Theorem 1.

(2) When $G_1(x) = x \oplus \alpha_1$ and $G_{K_{10}}(x) = x \dot{+} \beta_{\lfloor K_{10}/2 \rfloor}$, define $y = G_1(x) = x \oplus \alpha_1$ and set $\widetilde{F}(y) = G_{K_{10}} \circ \cdots \circ G_2(y)$. Since $F(x) = x \oplus \gamma$, then $\widetilde{F}(y) = x \oplus \alpha_1 \oplus \gamma = y \oplus (\alpha_1 \oplus \gamma)$. So, from Theorem 1, we have $\alpha_1 \oplus \gamma \in \left\{ \bigoplus_{i=2}^{\lfloor K_{10}/2 \rfloor} \alpha_i, \left(\bigoplus_{i=2}^{\lfloor K_{10}/2 \rfloor} \alpha_i \right) \oplus 128 \right\}$, which immediately leads to $\gamma \in \left\{ \bigoplus_{i=1}^{\lfloor K_{10}/2 \rfloor} \alpha_i, \left(\bigoplus_{i=1}^{\lfloor K_{10}/2 \rfloor} \alpha_i \right) \oplus 128 \right\}$.

(3) When $G_1(x) = x \dot{+} \beta_1$ and $G_{K_{10}}(x) = x \oplus \alpha_{\lfloor K_{10}/2 \rfloor}$, rewrite $F(x) = G_{K_{10}} \circ \widetilde{F}(x) = \widetilde{F}(x) \oplus \alpha_{\lfloor K_{10}/2 \rfloor}$, where $\widetilde{F}(x) = G_{K_{10}-1} \circ \cdots \circ G_1(x)$. From $F(x) = x \oplus \gamma$, one has $\widetilde{F}(x) = x \oplus (\gamma \oplus \alpha_{\lfloor K_{10}/2 \rfloor})$. Then, performing condition a) or b) on $\widetilde{F}(x)$, we have $\gamma \oplus \alpha_{\lfloor K_{10}/2-1 \rfloor} \in \left\{ \bigoplus_{i=1}^{\lfloor K_{10}/2-1 \rfloor} \alpha_i, \left(\bigoplus_{i=1}^{\lfloor K_{10}/2-1 \rfloor} \alpha_i \right) \oplus 128 \right\}$ and then get $\gamma \in \left\{ \bigoplus_{i=1}^{\lfloor K_{10}/2 \rfloor} \alpha_i, \left(\bigoplus_{i=1}^{\lfloor K_{10}/2 \rfloor} \alpha_i \right) \oplus 128 \right\}$.

■

Now let us try to find the answer to another question: how can we find encryption functions that are equivalent to $x \oplus \gamma$? From Proposition 1 and Corollary 2, this can be done by checking if the following 255 equalities hold: $F(x_1) \oplus F(x_1 \oplus i) = i$, where x_1 is an arbitrary integer in $\{0, \ldots, 255\}$ and $i = 1 \sim 255$.

Proposition 1 *Let $F(x) = y$ be a function defined over $\{0, \ldots, 255\}$ such that there exists $x_1 \in \{0, \ldots, 255\}$ verifying $F(x_1) \oplus F(x_1 \oplus i) = i, \forall i \in \{1, \ldots, 255\}$. Then $F(x) = x \oplus \gamma$ for some $\gamma \in \{0, \ldots, 255\}$ where $F(0) = \gamma$.*

Proof: Take any $x \in \{0, \ldots, 255\}$ and apply the hypothesis to $i = x \oplus x_1$ to obtain $F(x_1) \oplus F(x) = x \oplus x_1$. This implies $F(x) = x \oplus x_1 \oplus F(x_1)$. Thus $\gamma = x_1 \oplus F(x_1) = F(0)$. ■

Corollary 2 *Let $F(x) = y$ be a function defined over $\{0, \ldots, 255\}$. If there exists $x_1 \in \{0, \ldots, 255\}$ such that $F(x_1) \oplus F(x_1 \oplus i) = i$, $\forall i \in \{1, \ldots, 255\}$. Then, $\forall x_2, x_3 \in \{0, \ldots, 255\}$, $F(x_2) \oplus F(x_3) = x_2 \oplus x_3$.*

Proof: From Proposition 1, we have $F(x_2) \oplus F(x_3) = (x_2 \oplus F(0)) \oplus (x_3 \oplus F(0)) = x_2 \oplus x_3$. Thus the corollary is proved. ■

For the encryption functions $E_i(x)$ composed of the two basic kinds of functions, the above result can be further simplified. From Proposition 2, Corollary 3, it is enough to check the following 64 equalities: $F(x_1) \oplus F(x_1 \oplus d) = d$, where x_1 is an arbitrary integer in $\{0, \ldots, 255\}$ and d iterates over all elements of a set $\mathbb{S}$ satisfying Eq. (15).

$$\begin{aligned}&(\mathbb{S} \cup \{x \oplus 64 | \forall\, x \in \mathbb{S}\}) \supset \{1, \ldots, 64\}\\ &\#(\mathbb{S}) = 64.\end{aligned} \tag{15}$$

Proposition 2 *Consider any encryption function $E_i(x)$ $(i = 1 \sim 3)$ defined in Eqs. (4)∼(6). If there exists $x_1 \in \{0, \ldots, 255\}$ such that $E_i(x_1) \oplus E_i(x_1 \oplus d) = d$, $\forall d \in \{1, \ldots, 127\}$, then $E_i(x) = x \oplus E_i(0)$.*

Proof: From Fact 3, one has $E_i(x_1) \oplus E_i(x_1 \oplus 128) = 128$ and $E_i(x_1) \oplus E_i(x_1 \oplus j \oplus 128) = j \oplus 128$ for $j = 1 \sim 127$. This means that $E_i(x_1) \oplus E_i(x_1 \oplus j) = j$ holds $\forall j \in \{1, \ldots, 255\}$. Then, from Proposition 1, $E_i(x) = x \oplus E_i(0)$. ■

Proposition 3 *For a function $F(x)$ verifying the hypothesis of Theorem 1, any one of the following three statements:*

- *$F(0) \oplus F(d) = d$,*
- *$F(0) \oplus F(64) = 64$,*
- *$F(0) \oplus F(d \oplus 64) = d \oplus 64$,*

can be derived from the other two, where d is any integer within the set $\{1, \ldots, 63\}$.

Proof: Let $d = \sum_{j=0}^{7} d_j \cdot 2^j$. For convenience we write $F(0) = \gamma = \sum_{j=0}^{7} \gamma_j \cdot 2^j$. Let us derive the third statement from the first and second one. The other combinations are proved in a similar way.

The first statement may be written as $F(d) = d \oplus \gamma$ for any $d \in \{1, \ldots, 63\}$. Consider the octet $d = (0, 0, d_5, \ldots, d_0)$. It is clear that $F_6(d) = F_6(0) = \gamma_6$. This implies that, at the seventh coordinate, the number of carry bits produced by d and by 0 have the same parity p_6 and it is independent of d. Now consider the octet $d' = (0, 1, d_5, \ldots, d_0)$. For the definition of F, since the first coordinates to be evaluated are the least significant, it is clear that $F_j(d') = F_j(d) = d_j \oplus \gamma_j$ for any $j = 0 \sim 5$. For the seventh coordinate, since the parity or the carry bits produced by the previous coordinates does not depend on d, we may conclude that $F_6(d') = F_6(d) \oplus 1 = \gamma_6 \oplus 1$.

To check the eight coordinate, we need the second hypothesis. We may write it as $F(64) = 64 \oplus \gamma$. Since $F_7(64) = F_7(0) = \gamma_7$, we conclude that, at the eight coordinate, the carry bits produced by 64 and by 0 have the same parity (let us call it p_7). From the first statement we know that, the effect in the seventh coordinate of the carry bits of d, d', 0 and 64 is identical (defined through p_6). For that reason, the effect of the carry bits in the eight coordinate may be reduced to the value of the seventh coordinate of the evaluated octet. But, as was said before, this only depends on p_7. Thus, we may conclude that $F_7(d') = F_7(d) = F_7(0) = F_7(64) = \gamma_7$. ■

Corollary 3 *For the encryption $E_i(x)$ $(i = 1 \sim 3)$ shown in Eqs. (4)∼(6), if there exists $x_1 \in \{0, \ldots, 255\}$ such that $E_i(x_1) \oplus E_i(x_1 \oplus d) = d$, $\forall\ d \in \mathbb{S}$ satisfying Eq. (15), then $E_i(x) = x \oplus E_i(0)$.*

Proof: This corollary is the direct consequence of Propositions 2 and 3. ■

Now, let us discuss how to carry out the differential attack. We choose 65 different plain-images $\{I_l\}_{l=0}^{64}$. The size of the images is $M \times N$ and the pixel values of the image I_l for any $l = 0 \sim 64$ is $R_l(i) = G_l(i) = B_l(i) = l$, where $i = 1 \sim MN$.

With the chosen plain-images and the corresponding cipher-image $\{I_l^*\}_{l=0}^{64}$, we can find out the encryption functions that are equivalent to $x \oplus \gamma$ in the following way.

Consider the cipher image of the red channel of the first block in all 65 images. The encryption function E_1 is the same for all of them. So, we know $E_1(x)$ for $x = 0 \sim 64$. Does E_1 verify $E_1(x) = x \oplus \gamma$ for some γ? From the previous technical results, it is enough to check that $E_1(x) = x \oplus E_1(0)$ for $x = 0 \sim 64$. This argument is valid for any encryption function E_i and for any block.

Now iterate over all the blocks of the images to find, for the red channel, the encryption functions equivalent to $x \oplus \gamma$ and construct a two columns matrix $\mathbb{N}$ in the following way. If the n-th block verifies that the corresponding encryption function satisfies $E_1(x) = x \oplus \gamma_n$, then add to $\mathbb{N}$ the row (n, γ_n). The dimension of the matrix is $S \times 2$, where S is the total number of blocks with encryption function equivalent to $x \oplus \gamma$ for some γ.

Now, according to the notation introduced in section 3.1, consider any of the blocks $\{I^{(16)}(T \cdot k + j)\}_{k=0}^{N_T-1}$ or $\{I^{(16)}(T \cdot k + j + \frac{T}{2})\}_{k=0}^{N_T-1}$. If any of them produce an entrance in $\mathbb{N}$, say for instance row s, then the corresponding γ is $\mathbb{N}(s, 2)$ and belongs to the set $\widetilde{\mathbb{A}}_j$, where

$$\begin{aligned}
\widetilde{\mathbb{A}}_j =& \mathbb{A}_j \cup \{x \oplus 128 | x \in \mathbb{A}_j\},\\
=& \{0, \tilde{a}_0, \tilde{a}_1, \tilde{a}_0 \oplus \tilde{a}_1,\\
& 255, \tilde{a}_0 \oplus 255, \tilde{a}_1 \oplus 255, \tilde{a}_0 \oplus \tilde{a}_1 \oplus 255,\\
& 128, \tilde{a}_0 \oplus 128, \tilde{a}_1 \oplus 128, \tilde{a}_0 \oplus \tilde{a}_1 \oplus 128,\\
& 127, \tilde{a}_0 \oplus 127, \tilde{a}_1 \oplus 127, \tilde{a}_0 \oplus \tilde{a}_1 \oplus 127\},
\end{aligned} \tag{16}$$

$\mathbb{A}_j = \{0, \tilde{a}_0, \tilde{a}_1, \tilde{a}_0 \oplus \tilde{a}_1, 255, \tilde{a}_0 \oplus 255, \tilde{a}_1 \oplus 255, \tilde{a}_0 \oplus \tilde{a}_1 \oplus 255\}$, $\tilde{a}_0 = a_0 \dotplus j \cdot K_{10}$ and $\tilde{a}_1 = a_1 \dotplus j \cdot K_{10}$.

Now, let us analyze the set $\widetilde{\mathbb{A}}_j$ for a given j. The cardinality of $\widetilde{\mathbb{A}}_j$ depends on $\tilde{a}_0$, $\tilde{a}_1$, and can be classified as follows:

- $\#(\widetilde{\mathbb{A}}_j) = 4$. This is the case when $\tilde{a}_0, \tilde{a}_1 \in \{0, 255, 128, 127\}$.

- $\#(\widetilde{\mathbb{A}}_j) = 8$. This is the case when one of the following conditions hold:
  · $\tilde{a}_0 = \tilde{a}_1 \notin \{0, 255, 128, 127\}$;
  · $\tilde{a}_0 \in \{0, 255, 128, 127\}$, $\tilde{a}_1 \notin \{0, 255, 128, 127\}$;
  · $\tilde{a}_1 \in \{0, 255, 128, 127\}$, $\tilde{a}_0 \notin \{0, 255, 128, 127\}$.

- $\#(\widetilde{\mathbb{A}}_j) = 16$: For the rest of the cases.

Considering the family $\{\#(\widetilde{\mathbb{A}}_j)\}_{j=0}^{N_T-1}$, information about a_0 and a_1 can be obtained as follows:

- When $\#(\widetilde{\mathbb{A}}_j) = 4 \ \forall \ j = 0 \sim N_T - 1$ then $a_0, a_1 \in \{0, 255, 128, 127\}$ and $K_{10} \in \{0, 128\}$.
- When $\#(\widetilde{\mathbb{A}}_j) \leq 8 \ \forall \ j = 0 \sim N_T - 1$ then one of the following conditions hold:
  · $a_0 = a_1 \notin \{0, 255, 128, 127\}$;
  · $a_0 \in \{0, 255, 128, 127\}$, $a_1 \notin \{0, 255, 128, 127\}$ and $K_{10} \in \{0, 128\}$;
  · $a_0 \notin \{0, 255, 128, 127\}$, $a_1 \in \{0, 255, 128, 127\}$ and $K_{10} \in \{0, 128\}$
- When $\#(\widetilde{\mathbb{A}}_j) \notin \{4, 8\}$ for some j then the rest of the combinations hold.

The second column of $\mathbb{N}$ provides elements of different $\widetilde{\mathbb{A}}_j$. From the above analysis, one can see that the scope of a_0, a_1 and K_{10} can be guessed by observing the whole $\{\#(\widetilde{\mathbb{A}}_j)\}_{j=0}^{N_T-1}$. The procedure to obtain such information will be discussed shortly.

The estimation of the needed values is mainly based on the following fact. For j_0, j_1 such that $j_0 \neq j_1 (\text{mod } T)$ and $j_0 \neq j_1 (\text{mod } \frac{T}{2})$, $\widetilde{\mathbb{A}}_{j_0}$ and $\widetilde{\mathbb{A}}_{j_1}$ may have twelve different elements. [4] Since every set $\widetilde{\mathbb{A}}_j$ is closed with respect to XOR operation, the cardinality of a searched version of $\widetilde{\mathbb{A}}_{j_0}$ may exceed 16 if one element of $\widetilde{\mathbb{A}}_{j_1}$ is loaded in it and then all possible XOR operations between pairs of elements are performed.

Based on this point, we can search for the period of Eq. (7) T from $\mathbb{N}$ by the following steps.

Step 1. Set i with initial value $i = 8$ and define the set $\mathbb{M} = \{0, 255, 128, 127\}$. For the rest of the procedure, consider the member of $\mathbb{N}$, $\mathbb{N}(1, 1)$.
Step 2. Consider $T^* = 2^i$ as an estimation of T. For any $s = 1 \sim S$, add $\mathbb{N}(s, 2)$ to $\mathbb{M}$ if $(\mathbb{N}(s, 1) - \mathbb{N}(1, 1)) \bmod T^* = 0$.
Step 3. Carry out the XOR operation between any two elements of $\mathbb{M}$, and add the result into $\mathbb{M}$ if the value is not contained in $\mathbb{M}$.

[4] Note that there have only eight different elements when $(a_0 \dot{+} j_0 K_{10}) \oplus (a_1 \dot{+} j_0 K_{10}) = (a_0 \dot{+} j_1 K_{10}) \oplus (a_1 \dot{+} j_1 K_{10})$.

Step 4. If $\#(\mathbb{M}) \leq 16$, then go to Step 2 with $i = i - 1$; else stop the search.

Let us see how this algorithm works with an example. Assume that $T = 2^8$. Suppose that $\mathbb{N}(1,1) = T \cdot k + j_0$ for some natural number k. Then $\mathbb{N}(1,2) \in \widetilde{\mathbb{A}}_{j_0}$. In the first round, the only s to pass the test are those of the form $\mathbb{N}(1,1) + n \cdot 256$ for some natural number n. Since the encryption functions corresponding to those blocks are defined with the same keys, then $\mathbb{N}(s,2)$ has to belong to $\widetilde{\mathbb{A}}_{j_0}$. Thus, the cardinal of $\mathbb{M}$ after the first round has to be at most 16. It also could be 4 or 8.

At the second round, if s pass the test in Step 2 then $s = \mathbb{N}(1,1) + n \cdot 128$ for some natural number n. Thus, the keys corresponding to the block $I^{(16)}(\mathbb{N}(1,1))$ and $I^{(16)}(\mathbb{N}(s,1))$ have a gap of 128. Since $128 \in \mathbb{M}$, $\mathbb{N}(s,2)$ already belongs to $\widetilde{\mathbb{A}}_{j_0}$. So, after the second round, the cardinal of $\mathbb{M}$ is not greater than 16. Again, it also could be 4 or 8.

Now, in the third round, if s pass the test in Step 2 then $s = \mathbb{N}(1,1) + n \cdot 64$ for some natural number n. Thus, the keys corresponding to the block $I^{(16)}(\mathbb{N}(1,1))$ and $I^{(16)}(\mathbb{N}(s,1))$ have a gap of 64. For that reason, $\mathbb{N}(s,2)$ does not need to belong to $\widetilde{\mathbb{A}}_{j_0}$. So, after the third round the cardinal of $\mathbb{M}$ could be greater than 16. It also could be less or equal to 16: it depends of the values of $\mathbb{N}$.

Observe that the previous argument may be repeated for any period T. In any case, it is concluded that the validation condition $\#(\mathbb{M}) > 16$ will be reached only when $T^* \leq \frac{T}{4}$, i.e., $T \geq 4 \cdot T^*$.

Observe that if the second column of $\mathbb{N}$ has only a few different values and they are related to $\mathbb{N}(1,2)$ in the sense that all of them belong to the same set $\widetilde{\mathbb{A}}_j$, then no information may be obtained from this procedure.

Once T is determined, we can search for the values of K_{10}, a_0 and b_0 by the correlation between $\widetilde{\mathbb{A}}_{j_0}$ and $\widetilde{\mathbb{A}}_{j_1}$, which can be described with the following steps:

Step 1. Enumerate all possible values of K_{10} and take the first one. From Fact 1, for a given guessed period T, there are many suitable values for K_{10}. For instance, for $T = 256$, any odd number is a candidate for K_{10}.

Step 2. Recover two different sets $\widetilde{\mathbb{A}}_{j_0}$ and $\widetilde{\mathbb{A}}_{j_1}$. Let j_0 and j_1 be such that $\mathbb{N}(1,1) = T \cdot k_1 + j_0$ and $\mathbb{N}(s_1,1) = T \cdot k_2 + j_1$ for some natural numbers k_1 and k_2 and for $s_1 \in \{1, \ldots, S\}$ such that $j_0 \neq j_1 (\text{mod } T)$ and $j_0 \neq j_1 (\text{mod } \frac{T}{2})$. To recover the set $\widetilde{\mathbb{A}}_{j_0}$, find in the first column of $\mathbb{N}$ values s such that $\mathbb{N}(1,1) = \mathbb{N}(s,1) (\text{mod } T)$. With all such values generate the set of all XOR possible combinations and the numbers $\{0, 255, 128, 127\}$. The same procedure may be repeated for $\widetilde{\mathbb{A}}_{j_1}$.

Step 3. Enumerate all possible combination of a_{0,j_0}, a_{1,j_0} from $\widetilde{\mathbb{A}}_{j_0}$ and take the first one. Observing Eq. (16), one may see that there are $\binom{3}{2} \cdot 4 \cdot 4 = 48$ possible combinations of a_{0,j_0}, a_{1,j_0} when $a_{0,j_0}, a_{1,j_0} \notin \{0, 255, 128, 127\}$.

Step 4. Construct the set $\widehat{\mathbb{A}}_{j_1}$ as follows:

$$\begin{aligned}\widehat{\mathbb{A}}_{j_1} = \{&0, 255, \hat{a}_0, \hat{a}_1, \overline{\hat{a}_0}, \overline{\hat{a}_1}, \hat{a}_0 \oplus \hat{a}_1, \overline{\hat{a}_0} \oplus \hat{a}_1, 128, 127, \hat{a}_0 \oplus 128,\\ &\hat{a}_1 \oplus 128, \hat{a}_0 \oplus 127, \hat{a}_1 \oplus 127, \hat{a}_0 \oplus \hat{a}_1 \oplus 128, \hat{a}_0 \oplus \hat{a}_1 \oplus 127\},\end{aligned}$$

where $\hat{a}_0 = a_{0,j_0} \dotplus dif \cdot K_{10}$ and $\hat{a}_1 = a_{1,j_0} \dotplus dif \cdot K_{10}$ and $dif = j_1 - j_0$.

Step 5. If $\widehat{\mathbb{A}}_{j_1} = \widetilde{\mathbb{A}}_{j_1}$, it means that the corresponding estimation of (K_{10}, a_0, a_1) is reasonably good. So, it should be considered as a possible solution. Go to Step 6.

If $\widehat{\mathbb{A}}_{j_1} \neq \widetilde{\mathbb{A}}_{j_1}$, it means that $(K_{10}, a_{0,j_0}, a_{1,j_1})$ is a bad estimation for (K_{10}, a_0, a_1) so it should be discarded Go to Step 4 with the next pair (a_{0,j_0}, a_{1,j_1}) of the enumeration constructed in Step 3.

Step 6. Consider the next value of K_{10} in the enumeration constructed in Step 1. Then go to Step 4, starting at the first pair (a_{0,j_0}, a_{1,j_1}) of the enumeration constructed in Step 3.

The performance of the above search can be analyzed as follows. Given one combination of $(K_{10}, a_{0,j_0}, a_{1,j_0})$ passing the previous validation, the following variants would also pass it.

- $(K_{10}, a_{0,j_0} \oplus 128, a_{1,j_0})$, $(K_{10}, a_{0,j_0}, a_{1,j_0} \oplus 128)$ and $(K_{10}, a_{0,j_0} \oplus 128, a_{1,j_0} \oplus 128)$;
- $(c, a_{0,j_0}, a_{1,j_0})$, where $(c \cdot dif) \bmod 256 = ((K_{10} \cdot dif) \bmod 256) \oplus 128$;
- $(c, a_{0,j_0}, a_{1,j_0})$, where $(c \cdot dif) \bmod 256 = (K_{10} \cdot dif) \bmod 256$;
- $(c, a_{0,j_0}, a_{1,j_0})$, where $(c \cdot dif) \dotplus (K_{10} \cdot dif) = 0$.

Obviously, the first two types are due to the same reason, which is described in Fact 3. The last one is caused by equality $\overline{a_{i,j_0}} \dotplus (c \cdot dif) = \overline{a_{i,j_0} \dotplus (K_{10} \cdot dif)} = \overline{\hat{a}_i}$, $i \in \{0, 1\}$. The number of the searched results depends on K_{10} and dif. To minimize the searched results, one can choose dif satisfying i_0 is as small as possible, where $2^{i_0} | dif$ and $2^{i_0+1} \nmid dif$. The value of a_0, a_1 can be recovered from a_{0,j_0}, a_{1,j_0} by $a_0 = (a_{0,j_0} - j_0 \cdot K_{10}) \bmod 256$, $a_1 = (a_{1,j_0} - j_0 \cdot K_{10}) \bmod 256$.

To validate the feasibility of the above attack, we carry out a real attack with random selected secret key "$2A84BCF35D70664E4751$". First, we find $T^* = 64$, so $T = 256$. Then we get $\widetilde{\mathbb{A}}_{88} = \{0, 255, 203, 62, 52, 193, 245, 10, 128, 127, 75, 190, 180, 65, 117, 138\}$, and $\widetilde{\mathbb{A}}_{134} = \{0, 255, 89, 204, 166, 51, 149, 106, 128, 127, 217, 76, 38, 179, 21, 234\}$. The total combinations passing the validation are shown in Table 5.

Table 5
The possible combination of K_{10}, $a_{0,88}$ and $a_{1,88}$ from the search.

K_{10}	$a_{0,88}$	$a_{1,88}$	K_{10}	$a_{0,88}$	$a_{1,88}$	K_{10}	$a_{0,88}$	a_1	K_{10}	$a_{0,88}$	$a_{1,88}$
17	203	62	$\boxed{81}$	$\boxed{203}$	$\boxed{62}$	145	203	62	209	203	62
17	203	190	81	203	190	145	203	190	209	203	190
17	75	62	81	75	62	145	75	62	209	75	62
17	75	190	81	75	190	145	75	190	209	75	190
47	52	193	111	52	193	175	52	193	239	52	193
47	52	65	111	52	65	175	52	65	239	52	65
47	180	193	111	180	193	175	180	193	239	180	193
47	180	65	111	180	65	175	180	65	239	180	65

In this case, there are 32 possible combinations of (K_{10}, a_0, a_1) passing the validation. Only the framed one in Table 5 is correct.

As will be discussed later, the value of the ratio $\frac{S}{MN}$ has strong relation with K_{10}. In the above experiment, $S = 162$. We can guess $K_{10} \in \{81, 111\}$. In addition, the speed of the encryption function is sensitive to K_{10}, which helps the attacker to determine the value of K_{10} if the running time of encryption function can be detected.

So, we can get 16 possible value of (K_4, K_7). Similarly, the 16 possible values of (K_5, K_8), (K_6, K_9) can be obtained from the green and blue channel respectively.

To further determine the values of $K_4 \sim K_9$, we have to exhaustively guess Y_0 to check the coincidence between the cipher-image and the encryption result obtained by the searched candidate sub-keys. Considering 16 pixels of one channel is enough for validating the search and $f(x) = f(1-x)$, we can estimate the complexity of this search is about $O(2^{23} \cdot 32 + 32 \cdot 2) = O(2^{28})$. Once Y_0 is obtained, the value of $K_1 \sim K_3$ can be recovered from Eq. (3).

Since 64 pairs of differential chosen-images are relatively much, we study the probability of successfully detecting encryption functions that are equivalent to $x \oplus \gamma$ with a smaller number of differential chosen-images. For a number of random secret key, we set the differential values with all elements in $\mathbb{S}_j = \{1+2^{6-j} \cdot k\}_{k=0}^{(64/2^{6-j})-1}$, $j = 0 \sim 6$. For each secret key, the ratio $\frac{N_6}{N_j}$ is computed, where N_j, N_6 are the numbers of functions passing the detection when the differential values are set with all elements in $\mathbb{S}_j$ and $\mathbb{S}_6$ respectively. The results are shown in Fig. 12. Note that only encryption functions involving $x \dot{+} \beta$ are counted. From the experiment, we know $O(10)$ differential chosen images may be enough for the attack especially when K_{10} is not large.

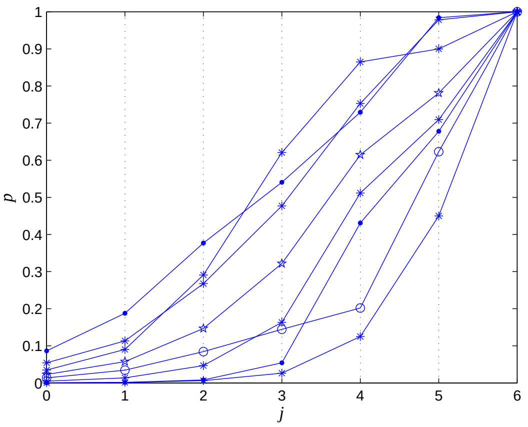


Fig. 12. Differential set $\mathbb{S}_j = \{1 + 2^{6-j} \cdot k\}_{k=0}^{(64/2^{6-j})-1}$ and corresponding probability of successful detection, $j = 0 \sim 6$.

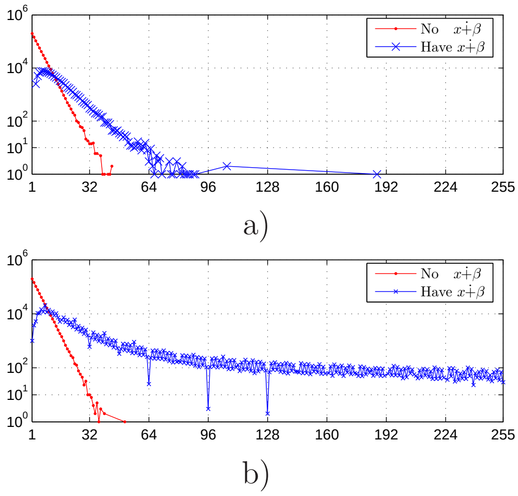
Finally, we discuss the feasibility of the differential attack proposed in this subsection. The most important point of the procedure is the recovering $\widetilde{\mathbb{A}}_{j_0}$ and $\widetilde{\mathbb{A}}_{j_1}$. In the better case, four encryption functions equivalent to $x \oplus \gamma$ should assure this point. However, the proposed attack is generally infeasible if the probability of an encryption function to be equal to $x \oplus \gamma$ is too small. When K_{10} is not very large, a lower bound of the probability can be estimated as the probability that an encryption function does not involve the second kind of sub-encryption-functions (i.e., functions of the form $x \dotplus \beta$). Assuming that the chaotic trajectory of the second Logistic map has an uniform distribution over $\{0, \ldots, 255\}$ and any two chaotic states are independent of each other, we can deduce that this lower bound is $(3/4)^{K_{10}}$. When K_{10} is relatively large, we can only turn to the encryption functions that are equivalent to $x \oplus \gamma$ even if the second kind of sub-encryption-functions are involved. From the proof of Theorem 1, we can see that being $E_i(x)$ equivalent to $x \oplus \gamma$ has a sensitive relation to $\{\alpha_{\lfloor j/2 \rfloor+1}\}_{j=1}^{len}$ and $\{\beta_{\lfloor j/2 \rfloor+1}\}_{j=1}^{len}$, in the sense that even one bit change of $\alpha_{\lfloor j/2 \rfloor+1}$ or $\beta_{\lfloor j/2 \rfloor+1}$ could make the equivalence to fail.

As a reference, we carried out a 1,000 times random experiment under some values of K_{10}, where sub-keys $K_1 \sim K_9$ are chosen randomly. The results are shown in Table 6, where the numbers of different sub-key sets that have at least one pixel satisfying $E_1(x) = x \oplus \gamma$, the mean value and standard variance of the sequence including the number of pixels satisfying the condition are shown from the first row to the third one respectively.

For a plain-image of size 512 × 512, a number of random experiments have been made with random selected $K_1 \sim K_9$ under different value of K_{10}. Two of such examples are shown in Fig. 13. In the figure, the encryption functions involving the second kind of sub-encryption-functions or not involving are counted respectively.

Table 6
Experiment results about the probability $E_1(x) = x \oplus \gamma$ under some values of K_{10}.

K_{10}	45	65	100	150	200	255
number	1000	1000	572	445	197	230
mean	324	146	194	238	775	329
variance	629	584	1408	1355	4486	1136



a)

b)

Fig. 13. The number of pixels satisfying $E_1(x) = x \oplus \gamma$ under different value of K_{10}: a) $K_1 \sim K_9$ = "$8DB87A1613D75ADF2D$"; b) $K_1 \sim K_9$ = "$2A84BCF35$ $D70664347$".

4 Conclusion

In this paper, the security of a recently proposed image encryption scheme has been studied in detail. It is found that there exist some serious problems with secret keys including invalid ones, weak ones and partially equivalent ones. The distribution of sub-images of the cipher-images is not uniform enough. A subkey even can be guessed from cipher-image of a chosen plain-image. Moreover, seven sub-keys among the ten ones can be recovered with a differential attack in the case that any 64 chosen plain-images satisfying a constraint is enough. The cryptanalysis presented in this paper also provide a thought for attacking schemes composing of multiple round encryption functions.

Acknowledgements

This research was partially supported by ...

References

[1] C. Alexopoulos, N. G. Bourbakis, N. Ioannou, Image encryption method using a class of fractals, J. Electronic Imaging 4 (3) (1995) 251–259.

[2] T.-J. Chuang, J.-C. Lin, New approach to image encryption, J. Electronic Imaging 7 (2) (1998) 350–356.

[3] J.-I. Guo, J.-C. Yen, H.-F. Pai, New voice over Internet protocol technique with hierarchical data security protection 149 (4) (2002) 237–243.

[4] H.-C. Chen, J.-C. Yen, A new cryptography system and its VLSI realization, J. Systems Architecture 49 (2003) 355–367.

[5] K.-L. Chung, L.-C. Chang, Large encryption binary images with higher security 19 (5–6) (1998) 461–468.

[6] Y. Mao, G. Chen, S. Lian, A novel fast image encryption scheme based on 3D chaotic Baker maps, Int. J. Bifurcation and Chaos 14 (10) (2004) 3613–3624.

[7] G. Chen, Y. Mao, C. K. Chui, A symmetric image encryption scheme based on 3d chaotic cat maps, Chaos, Solitons & Fractals 21 (3) (2004) 749–761.

[8] S. Li, C. Li, G. Chen, D. Zhang, N. G. Bourbakis, A general cryptanalysis of permutation-only multimedia encryption algorithms, IACR's Cryptology ePrint Archive: Report 2004/374, available at `eprint.iacr.org/2004/374` (2004).

[9] S. Li, C. Li, G. Chen, X. Mou, Cryptanalysis of the RCES/RSES image encryption scheme, IACR's Cryptology ePrint Archive: Report 2004/376, available online at `http://eprint.iacr.org/2004/376` (2004).

[10] C. Li, S. Li, D. Zhang, G. Chen, Cryptanalysis of a chaotic neural network based multimedia encryption scheme, Lecture Notes in Computer Science **3333**, Springer-Verlag, 2004, pp. 418–425.

[11] C. Li, S. Li, K.-T. Lo, G. Chen, Cryptanalysis of an image encryption scheme, arXiv e-print, cs.CR/0608024, available at `http://www.arxiv.org/pdf/cs.CR/0608024` (2006).

[12] C. Li, S. Li, D. Zhang, G. Chen, Cryptanalysis of a data security protection scheme for VoIP, IEE Proceedings – Vision, Image & Signal Processing 153 (1) (2006) 1–10.

[13] B. Furht, D. Socek, A. M. Eskicioglu, Fundamentals of multimedia encryption techniques, in: B. Furht, D. Kirovski (Eds.), Multimedia Security Handbook, CRC Press, 2004, Ch. 3, pp. 93–132.

[14] S. Li, G. Chen, X. Zheng, Chaos-based encryption for digital images and videos, in: B. Furht, D. Kirovski (Eds.), Multimedia Security Handbook, CRC Press, 2004, Ch. 4, pp. 133–167, preprint is available at `http://www.hooklee.com/pub.html`.

[15] A. Uhl, A. Pommer, Image and Video Encryption: From Digital Rights Management to Secured Personal Communication, Springer Science + Business Media Inc., Boston, 2005.

[16] N. Pareek, V. Patidar, K. Sud, Discrete chaotic cryptography using external key, Physics Letters A 309 (1-2) (2003) 75–82.

[17] N. Pareek, V. Patidar, K. Sud, Cryptography using multiple one-dimensional chaotic maps, Communications in Nonlinear Science and Numerical Simulation 10 (7) (2005) 715–723.

[18] N. Pareek, V. Patidar, K. Sud, Image encryption using chaotic logistic map, Image and Vision Computing 24 (9) (2006) 926–934.

[19] G. Álvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of a discrete chaotic cryptosystem using external key, Physics Letters A 319 (3-4) (2003) 334–339.

[20] C. Li, S. Li, G. Álvarez, G. Chen, K.-T. Lo, Cryptanalysis of a chaotic block cipher with external key and its improved version, Chaos, Solitons & Fractals, in press, doi:10.1016/j.chaos.2006.08.025 (2006).