Breaking a chaotic image encryption algorithm based on modulo addition and XOR operation

Chengqing Li[1],[*] Yuansheng Liu[1], Leo Yu Zhang[2], and Michael Z. Q. Chen[3]

[1] College of Information Engineering,
Xiangtan University, Xiangtan 411105, Hunan, China

[2] School of Mathematics and Computational Science,
Xiangtan University, Xiangtan 411105, Hunan, China

[3] Department of Mechanical Engineering,
The University of Hong Kong, Hong Kong

This paper re-evaluates the security of a chaotic image encryption algorithm called MCKBA/HCKBA and finds that it can be broken efficiently with two known plain-images and the corresponding cipher-images. In addition, it is reported that a previously proposed breaking on MCKBA/HCKBA can be further improved by reducing the number of chosen plain-images from four to two. The two attacks are both based on the properties of solving a composite function involving the carry bit, which is composed of the modulo addition and the bitwise OR operations. Both rigorous theoretical analysis and detailed experimental results are provided.

Keywords: image encryption; chaos; chosen-plaintext attack; carry bit.

1. Introduction

The subtle similarities between chaos and cryptography make chaos considered as a special way to design secure and efficient encryption schemes [Chen *et al.*, 2004, 2011]. Meanwhile, some cryptanalysis work demonstrated that some chaos-based encryption schemes are vulnerable to various conventional attacks from the viewpoint of modern cryptology [Li *et al.*, 2004; Xiao *et al.*, 2006; Solak *et al.*, 2010a,b]. In addition, some specific security flaws of chaos-based encryption schemes were reported [Zhou & Au, 2011; Chen *et al.*, 2012]. [Álvarez & Li, 2006] concluded some general approaches to evaluating security of chaos-based encryption schemes.

Due to the simplicity and low computation complexity of bitwise exclusive OR operation and modulo addition, they are widely used in traditional text encryption schemes and hash functions. Possible generation of carry bit by the modulo addition makes the two operations are neither identical nor interchangeable. Some properties existing in multi-round combination of the two basic operations were derived to facilitate differential attacks on some traditional text encryption schemes or searching collision of hash functions [Paul & Preneel, 2005; Wang *et al.*, 2005]. Among many chaos-based encryption schemes, the two operations are the basic involved (even only) substitution functions. In [Li *et al.*, 2005], [Li *et al.*, 2006], [Li *et al.*, 2008], and [Li *et al.*, 2009], the following properties about n-bit integers $\alpha, \beta, \gamma, x, y$ were found to support or enhance the proposed attacks on the corresponding encryption schemes in turn.

- If $\alpha \oplus \beta = 2^n - 1$, then equation $(\alpha \oplus x) = (\beta \oplus x) \dotplus \gamma$ has unique solution modulo 2^{n-1}, where $(a \dotplus b) = (a + b) \bmod 2^n$;
- Equation $|(\alpha \oplus \beta) - (\beta \oplus \gamma)| = |(\alpha \oplus \bar{\beta}) - (\beta \oplus \bar{\gamma})|$ always exists;
- If $\alpha \oplus \beta = \gamma$, then $|\alpha - \beta| \leq \gamma$;
- If $(((x \dotplus \alpha) \oplus \beta) \dotplus \gamma) \equiv x \oplus y$, then $y \equiv \beta \pmod{2^{n-1}}$.

In 2000, Yen *et al.* proposed a chaotic key-based algorithm (CKBA) by encrypting each pixel of a plain-image by four possible operations: XORing or XNORing it with one of two predefined sub-keys. The exerted operation is determined by a pseudo-random number sequence (PRNS) generated by iterating the logistic map [Yen & Guo, 2000]. In 2002, S. Li *et al.* broke CKBA with only one known/chosen-image in [Li & Zheng, 2002]. In 2005, Socek *et al.* proposed an enhanced version of CKBA (ECKBA) employing the following four methods: 1) replacing the logistic map with a piecewise linear chaotic map (PWLCM); 2) increasing the bit length of secret key to 128; 3) adding a modulo addition and an XOR operation; 4) running all the basic encryption functions multiple times. To achieve a much better balance between encryption load and security of high level, in 2007 Rao *et al.* proposed a modified version of CKBA (MCKBA) in [Rao & Gangadhar, 2007] by employing a modular addition operation like [Socek *et al.*, 2005]. To further enhance the security of MCKBA against brute-force attack, in 2010 Gangadhar *et al.* replaced the logistic map with a simple hyperchaos generator proposed in [Takahashi *et al.*, 2004] and names the algorithm HCKBA (Hyper Chaotic-Key Based Algorithm) [Gangadhar & Rao, 2010]. Since the two schemes MCKBA and HCKBA share the same structure, [Li *et al.*, 2011] analyzed them together and reported the following points:

- Equivalent secret key of MCKBA/HCKBA can be obtained from four pairs of chosen-plaintexts;
- Encryption result of MCKBA/HCKBA is not sensitive to changes of plain-image;
- Encryption result of MCKBA is not sensitive to changes of two sub-keys.
- The lower bound on the number of queries (α, β) to solve unknown variable x in equation

$$y = (\alpha \dotplus x) \oplus (\beta \dotplus x) \tag{1}$$

in terms of modulo 2^{n-1} is 3 if $n \geq 4$.

This paper re-evaluates the security of MCKBA/HCKBA and reports the following points: 1) some properties of Eq. (1) are provided to support practical approaches to solving Eq. (1); 2) MCKBA/HCKBA can be efficiently broken with two known-plaintexts; 3) the chosen-plaintext attack proposed in [Li *et al.*, 2011] can be further improved and the number of required chosen-plaintexts is only two.

The rest of this paper is organized as follows. The image encryption algorithm under study is briefly introduced in Sec. 2. A known-plaintext attack and an improved chosen-plaintext attack on the algorithm is presented in Sec. 3 with experimental results. The last section concludes this paper.

2. The Chaotic Image Encryption Algorithm Under Study

The encryption object of MCKBA is a gray-scale image of size $M \times N$ (width$\times$height), which is scanned in the raster order and represented as a one dimensional sequence $\boldsymbol{I} = \{I(i)\}_{i=0}^{MN-1}$. Then, a binary sequence $\boldsymbol{I_b} = \{I_b(l)\}_{l=0}^{8MN-1}$ is constructed, where $\sum_{j=0}^{7} I_b(8 \cdot i + j) \cdot 2^j = I(i)$ for $i = 0 \sim MN - 1$. With a pre-defined integer parameter n, an n-bit number sequence $\boldsymbol{J} = \{J(k)\}_{k=0}^{\lceil 8MN/n \rceil - 1}$ is generated, where $J(k) = \sum_{j=0}^{n-1} I_b(n \cdot k + j) \cdot 2^j$. In case $(8MN)$ is not a multiple of n, the sequence $\boldsymbol{I_b}$ is padded with some zero bits. Without loss of generality, it is assumed that n can divide $(8MN)$ in this paper. MCKBA operates on the intermediate sequence $\boldsymbol{J}$ and obtains $\boldsymbol{J'} = \{J'(k)\}_{k=0}^{8MN/n-1}$, where $J'(k) = \sum_{j=0}^{n-1} I_b'(n \cdot k + j) \cdot 2^j$. Finally, cipher-image $\boldsymbol{I'} = \{I'(k)\}_{k=0}^{MN-1}$ is obtained via $I'(k) = \sum_{j=0}^{7} I_b'(8 \cdot k + j) \cdot 2^j$. Based on the above preliminary introduction, MCKBA is described with the following four parts[1].

---

[1]Since the sole difference between MCKBA and HCKBA is the generator of PRBS, only MCKBA is introduced here with a concise and consistent form to illustrate the encryption procedure.

- *The secret key*: Two random numbers key_1, $key_2 \in \{0, \cdots, 2^n - 1\}$, and the initial condition $x(0) \in (0, 1)$ of the logistic map

$$x(k + 1) = 3.9 \cdot x(k) \cdot (1 - x(k)), \tag{2}$$

where $\sum_{j=0}^{n-1}(key_{1,j} \oplus key_{2,j}) = \lceil n/2 \rceil$, $key_1 = \sum_{j=0}^{n-1} key_{1,j} \cdot 2^j$, $key_2 = \sum_{j=0}^{n-1} key_{2,j} \cdot 2^j$, and $\oplus$ denotes the eXclusive OR (XOR) operation.

- *Initialization*: Run Eq. (2) iteratively to generate a sequence $\{x(k)\}_{k=0}^{MN/(2n)-1}$ and derive a pseudo-random binary sequence (PRBS), $\{b(l)\}_{l=0}^{16MN/n-1}$, from the 32-bit binary representation of elements of the sequence, namely $x(k) = \sum_{j=1}^{32} b(32 \cdot k + j - 1) \cdot 2^{-j}$.

- *Encryption*: For $k = 0 \sim 8MN/n - 1$, encrypt the k-th plain-element of $\boldsymbol{J}$ via

$$J'(k) = \begin{cases} (J(k) \dotplus key_1) \oplus key_1 & \text{if } B(k) = 3; \\ (J(k) \dotplus key_1) \odot key_1 & \text{if } B(k) = 2; \\ (J(k) \dotplus key_2) \oplus key_2 & \text{if } B(k) = 1; \\ (J(k) \dotplus key_2) \odot key_2 & \text{if } B(k) = 0, \end{cases} \tag{3}$$

where $B(k) = 2 \cdot b(2k) + b(2k + 1)$, and $a \odot b = \overline{a \oplus b} = a \oplus \bar{b}$.

- *Decryption*: The decryption procedure is similar to that of the encryption except that Eq. (3) is replaced by

$$J(k) = \begin{cases} (J'(k) \oplus key_1) \dotminus key_1 & \text{if } B(k) = 3; \\ (J'(k) \oplus \overline{key_1}) \dotminus key_1 & \text{if } B(k) = 2; \\ (J'(k) \oplus key_2) \dotminus key_2 & \text{if } B(k) = 1; \\ (J'(k) \oplus \overline{key_2}) \dotminus key_2 & \text{if } B(k) = 0, \end{cases} \tag{4}$$

where $a \dotminus b = (a - b + 2^n) \bmod 2^n$.

3. Cryptanalysis

Assume that two plain-images and the corresponding cipher-images encrypted with the same secret key are available, and let $\boldsymbol{J}_1 = \{J_1(k)\}_{k=0}^{8MN/n-1}$ and $\boldsymbol{J}_2 = \{J_2(k)\}_{k=0}^{8MN/n-1}$ denote the corresponding intermediate sequences, respectively. Then, one can assure that the two sequences and the corresponding encrypted results $\boldsymbol{J}_1' = \{J_1'(k)\}_{k=0}^{8MN/n-1}$ and $\boldsymbol{J}_2' = \{J_2'(k)\}_{k=0}^{8MN/n-1}$ satisfy

$$J_1'(k) \oplus J_2'(k) = \begin{cases} (J_1(k) \dotplus key_1) \oplus (J_2(k) \dotplus key_1) & \text{if } B(k) \in \{2, 3\}; \\ (J_1(k) \dotplus key_2) \oplus (J_2(k) \dotplus key_2) & \text{if } B(k) \in \{0, 1\}. \end{cases} \tag{5}$$

No matter what the value of $B(k)$ is, the above equation can be represented in the form of Eq. (1). In this section, we first present some properties of the kernel function (1) on obtaining its solution and then illustrate how to obtain an equivalent secret key of MCKBA/HCKBA with two known plain-images and two chosen plain-images, respectively.

3.1. *Some properties of the kernel function*

Property 1. Equivalent form of Eq. (1)

$$\tilde{y} = y \oplus \alpha \oplus \beta = (\alpha \dotplus x) \oplus (\beta \dotplus x) \oplus \alpha \oplus \beta \tag{6}$$

can be represented as an iteration form

$$\begin{cases} \tilde{y}_{i+1} = c_{i+1} \oplus \tilde{c}_{i+1}, \\ c_{i+1} = (x_i \cdot \alpha_i) \oplus (x_i \cdot c_i) \oplus (\alpha_i \cdot c_i), \\ \tilde{c}_{i+1} = (x_i \cdot \beta_i) \oplus (x_i \cdot \tilde{c}_i) \oplus (\beta_i \cdot \tilde{c}_i), \end{cases} \tag{7}$$

where $c_0 \equiv 0$, $\tilde{c}_0 \equiv 0$, $x = \sum_{i=0}^{n-1} x_i \cdot 2^i$, $\alpha = \sum_{i=0}^{n-1} \alpha_i \cdot 2^i$, $\beta = \sum_{i=0}^{n-1} \beta_i \cdot 2^i$, $\tilde{y} = \sum_{i=0}^{n-1} \tilde{y}_i \cdot 2^i$ (These notations are the same hereinafter.).

Proof. Set $c_0 = 0$, one can calculate c_{i+1} from c_i and α_i via

$$c_{i+1} = (x_i \cdot \alpha_i) \oplus (x_i \cdot c_i) \oplus (\alpha_i \cdot c_i) \tag{8}$$

for $i = 0 \sim n - 2$. So, c_{i+1} denote the carry bit generated by x and α in the i-th bit plane. Set $\tilde{c}_0 = 0$, one can then obtain

$$\tilde{c}_{i+1} = (x_i \cdot \beta_i) \oplus (x_i \cdot \tilde{c}_i) \oplus (\beta_i \cdot \tilde{c}_i)$$

for $i = 0 \sim n - 2$. Similarly, $\tilde{c}_{i+1}$ denote the carry bit generated by x and β in the i-th bit plane. Obviously, $\tilde{y}_0 = (\alpha_0 \oplus x_0) \oplus (\beta_0 \oplus x_0) \oplus \alpha_0 \oplus \beta_0 \equiv 0$. Then, the $(i+1)$-th bit plane of Eq. (6) can be represented as

$$\tilde{y}_{i+1} = (\alpha_{i+1} \oplus c_{i+1} \oplus x_{i+1}) \oplus (\beta_{i+1} \oplus \tilde{c}_{i+1} \oplus x_{i+1}) \oplus \alpha_{i+1} \oplus \beta_{i+1}$$
$$= c_{i+1} \oplus \tilde{c}_{i+1},$$

where $i = 0 \sim n - 2$. So, $\tilde{y}_i$ can be easily calculated iteratively according to Eq. (7) for $i = 0 \sim n - 2$, which can also be done via checking Table 1 listing the values of $\tilde{y}_{i+1}$ under all possible different values of $\alpha_i, \beta_i, \tilde{y}_i, x_i$, and c_i.   ∎

Table 1.   The values of $\tilde{y}_{i+1}$ corresponding to the values of $\alpha_i, \beta_i, \tilde{y}_i, x_i$, and c_i.

| (x_i, c_i) | $(\alpha_i, \beta_i, \tilde{y}_i)$ | | | | | | | |
	$(0,0,0)$	$(0,0,1)$	$(0,1,0)$	$(0,1,1)$	$(1,0,0)$	$(1,0,1)$	$(1,1,0)$	$(1,1,1)$
$(0,0)$	0	0	0	1	0	0	0	1
$(0,1)$	0	0	1	0	1	1	0	1
$(1,0)$	0	1	1	1	1	0	0	0
$(1,1)$	0	1	0	0	0	1	0	0

Property 2. Given $(\alpha_i, \beta_i, \tilde{y}_i, \tilde{y}_{i+1})$, no information about x_i, c_i and $\tilde{c}_i$ can be obtained (Note that c_0 and $\tilde{c}_0$ are excluded since they are pre-defined constants.) if and only if $(4\alpha_i + 2\beta_i + \tilde{y}_i) \in \{0, 6\}$.

Proof. Since only the data in the 0, 6-th column (zero-based) of Table 1 are identical, it is impossible to obtain any information about x_i, c_i and $\tilde{c}_i$ from $(\alpha_i, \beta_i, \tilde{y}_i, \tilde{y}_{i+1})$ if and only if $(4\alpha_i + 2\beta_i + \tilde{y}_i) \in \{0, 6\}$.   ∎

Property 3. Given $(\alpha_i, \beta_i, \tilde{y}_i, \tilde{y}_{i+1})$, the unknown bit x_i can be determined via $x_i = \alpha_i \oplus \tilde{y}_{i+1}$, if and only if $(4\alpha_i + 2\beta_i + \tilde{y}_i) \in \{1, 7\}$.

Proof. Only under the cases shown in the 1, 7-th column of Table 1, x_i can be determined by $(\alpha_i, \beta_i, \tilde{y}_i, \tilde{y}_{i+1})$ without knowledge of c_i. It is easy to verify that $x_i = \alpha_i \oplus \tilde{y}_{i+1}$ in terms of value.   ∎

Property 4. Given $(\alpha_i, \beta_i, \tilde{y}_i, \tilde{y}_{i+1})$, carry bits c_i and $\tilde{c}_i$ can be determined via $c_i = \beta_i \oplus \tilde{y}_{i+1}$ and $\tilde{c}_i = \beta_i \oplus \tilde{y}_{i+1} \oplus \tilde{y}_i$ if and only if $(4\alpha_i + 2\beta_i + \tilde{y}_i) \in \{3, 5\}$.

Proof. Only under the cases shown in the 3, 5-th column of Table 1, c_i can be determined by $(\alpha_i, \beta_i, \tilde{y}_i, \tilde{y}_{i+1})$ without knowledge of x_i. It is easy to verify that $c_i = \beta_i \oplus \tilde{y}_{i+1}$ in terms of value. Then, one can obtain $\tilde{c}_i = \beta_i \oplus \tilde{y}_{i+1} \oplus \tilde{y}_i$ since $\tilde{c}_i = c_i \oplus \tilde{y}_i$.   ∎

Property 5. Given $(\alpha_i, \beta_i, \tilde{y}_i, \tilde{y}_{i+1})$, the scope of the unknown bits x_i, c_i can be narrowed via

$$(x_i, c_i) \in \begin{cases} \{(0,0), (1,1)\} & \text{if } \tilde{y}_{i+1} = 0; \\ \{(0,1), (1,0)\} & \text{if } \tilde{y}_{i+1} = 1, \end{cases} \tag{9}$$

if and only if $(4\alpha_i + 2\beta_i + \tilde{y}_i) \in \{2, 4\}$.

Proof. Referring to the cases shown in the 2, 4-th column of Table 1, the scope of (x_i, c_i) can be narrowed according to value of $\tilde{y}_{i+1}$. It is easy to obtain Eq. (9) from Table 1. Therefore, the "if" part of the property is proven. Note that the number of possible values of $(4\alpha_i + 2\beta_i + \tilde{y}_i)$ is only eight, and the sufficient and necessary conditions on obtaining different information on x_i and c_i under other six cases have been presented. Therefore, the "only if" part of the property is also proven. ■

Property 6. If $(\alpha_{i-1} \oplus \beta_{i-1}) = 1$ and $\tilde{y}_i = 1$, one has $c_i = \alpha_{i-1}$, where $i \in \{1, 2, \cdots, n-2\}$.

Proof. When $(\alpha_{i-1}, \beta_{i-1}) = (1, 0)$ and $\tilde{y}_i = 1$, one can get $c_i = 1$ from Eq. (7) no matter the value of x_{i-1}. Similarly, one has $c_i = 0$ when $(\alpha_{i-1}, \beta_{i-1}) = (0, 1)$ and $\tilde{y}_i = 1$. So, the property is proven. ■

Property 7. Given $\alpha, \beta, \tilde{y}$, some bits among the $(n-1)$ least significant bits of x in Eq. (6) can be determined from the least significant bit to the most significant one.

Proof. The concrete approaches to solving Eq. (1) and determining the carry bits can be divided into the following two classes of operations.

- *Obtaining information on x_0 and c_1*: According to how much information on x_0 and c_1 can be obtained, $(\alpha_0, \beta_0, \tilde{y}_0)$ is further classified as the following two cases.

(a) $(4\alpha_0 + 2\beta_0 + \tilde{y}_0) \in \{0, 6\}$: Referring to Property 2, x_0 can not be determined in this case, but one can obtain $c_1 = 0$ if $\alpha_0 = 0$.

(b) $(4\alpha_0 + 2\beta_0 + \tilde{y}_0) \in \{2, 4\}$: As $c_0 = 0$, one can obtain

$$x_0 = \begin{cases} 0 & \text{if } \tilde{y}_1 = 0; \\ 1 & \text{if } \tilde{y}_1 = 1, \end{cases} \tag{10}$$

from Eq. (9). Then, one can further obtain

$$c_1 = \begin{cases} 0 & \text{if } \tilde{y}_1 = 0; \\ 1 & \text{if } \alpha_0 = 1 \text{ and } \tilde{y}_1 = 1; \\ 0 & \text{if } \alpha_0 = 0 \text{ and } \tilde{y}_1 = 1. \end{cases}$$

- *Obtaining information on x_i, c_i and c_{i+1} for $i = 1 \sim n - 2$*: According to how much information on x_i, c_i and c_{i+1} can be obtained by checking $(\alpha_i, \beta_i, \tilde{y}_i)$ and the obtained information on c_i for $i = 1 \sim n - 2$ in order, $(\alpha_i, \beta_i, \tilde{y}_i)$ is categorized as the following four cases[2].

(a) $(4\alpha_i + 2\beta_i + \tilde{y}_i) \in \{0, 6\}$: Referring to Property 2, no information on x can be determined in this case. The value of c_{i+1} can be determined by Eq. (8) if

$$\{c_i + \alpha_i\} \in \{0, 2\} \text{ is known.} \tag{11}$$

(b) $(4\alpha_i + 2\beta_i + \tilde{y}_i) \in \{1, 7\}$: One has

$$x_i = \alpha_i \oplus \tilde{y}_{i+1} \tag{12}$$

from Property 3. If c_i has been determined, one can obtain c_{i+1}. Even c_i is still unknown, one can confirm c_{i+1} by Eq. (8) if $(\alpha_i + x_i) = 0$ or $(\alpha_i + x_i) = 2$ is known.

(c) $(4\alpha_i + 2\beta_i + \tilde{y}_i) \in \{2, 4\}$: If c_i has been determined, based on Property 5 one can obtain

$$x_i = \begin{cases} 1 - \tilde{y}_{i+1} & \text{if } c_i = 1; \\ \tilde{y}_{i+1} & \text{if } c_i = 0, \end{cases} \tag{13}$$

and further confirm the value of c_{i+1}.

(d) $(4\alpha_i + 2\beta_i + \tilde{y}_i) \in \{3, 5\}$: Referring to Property 4, one can obtain $c_i = \beta_i \oplus \tilde{y}_{i+1}$.

■

---

[2] As confirmation of c_i is equivalent to that of $\tilde{c}_i$, the latter is not mentioned.

3.2. *Known-plaintext attack*

Known-plaintext attack is one of the classic attack models where the attacker (or cryptanalyst) can access both some plaintexts and the corresponding encryption results encrypted with the same secret key. In [Gangadhar & Rao, 2010, Sec. 3.2], the original authors claimed that HCKBA has strong vulnerability against known-plaintext attack. However, we found MCKBA/HCKBA is very weak against the attack, which is supported by the properties of Eq. (1) shown in the previous subsection.

Under the scenario of known-plaintext attack, breaking MCKBA/HCKBA is to determine some information of its equivalent secret key, $key1$, $key2$ and $\{B(k)\}_{k=0}^{8MN/n-1}$, by solving Eq. (5) and utilizing some properties of MCKBA/HCKBA. From Property 7, one can see that some bits of $key1$ and $key2$ can be obtained from Eq. (5) for any $k \in \{0, \cdots, 8MN/n - 1\}$, where the other unknown bits are just set as zero. Let $key(k)$ denote the obtained solution of Eq. (5) and $s(k, i)$ represent $key(k)_i$ is confirmed definitely or not, i.e. set $s(k, i) = 1$ if $key(k)_i$ is confirmed by Eq. (10), Eq. (12), or Eq. (13); otherwise set $s(k, i) = 0$, where $key(k) = \sum_{i=0}^{n-1} key(k)_i \cdot 2^i$, and $k = 0 \sim 8MN/n - 1$. Then, one may reconstruct set $\{key1, key2\}$ from $\{key(k)\}_{k=0}^{8MN/n-1}$ and $\{s(k, i)\}_{k=0,i=0}^{8MN/n-1,n-2}$ by identifying and combining the known bits belonging to the same number, which is described by the following steps.

- *Step 1)*: Set $\mathbb{K} = \{key(0), key(1), \cdots, key(8MN/n-1)\}$. Delete some elements of $\mathbb{K}$ to assure that every pair of elements of $\mathbb{K}$ has at least one different confirmed bit.
- *Step 2)*: Search for the first two elements in $\mathbb{K}$ whose number of confirmed bits are most but the confirmed bits of the two elements are not all the same. Let $Seed(0)$ and $Seed(1)$ denote the two seed elements and delete them from $\mathbb{K}$.
- *Step 3)*: Check each element of $\mathbb{K}$ in turn and do the following two operations if it has one confirmed bit which is different from that of $Seed(i)$: 1) update $Seed(1-i)$ by combining all the confirmed bits of the element into that of $Seed(1-i)$; 2) delete the element from $\mathbb{K}$, where $i \in \{0, 1\}$.
- *Step 4)*: Repeat *Step 3)* iteratively till the numbers of confirmed bits of $Seed(0)$ and $Seed(1)$ are not increased in the whole step.
- *Step 5)*: Terminate the whole search operation when all bits of $Seed(0)$ and $Seed(1)$ are confirmed bits; otherwise repeat *Step 2)* through *Step 4)* till the cardinality of $\mathbb{K}$ is less than 2.

Let us study the probability on obtaining x_i and c_i with one pair of α, β and $\tilde{y}$ under assumption that α, β and x distributes over $\{0, \cdots, 2^n - 1\}$ uniformly. First, one has $Prob(c_0 = 1) = 0$ and

$$Prob(c_i = 1) = \frac{3}{4}Prob(c_{i-1} = 1) + \frac{1}{4}Prob(c_{i-1} = 0)$$

for $i = 1 \sim n - 1$. Solve the above iteration function, one can obtain $Prob(c_i = 1) = \frac{2^i - 1}{2^{i+1}}$. Obviously, one has $Prob(\tilde{y}_0 = 0) = 1$. Observe Table 1, one can see that the value of $\tilde{y}_i$ is determined by the values of $\tilde{y}_{i-1}$, c_{i-1}, x_{i-1}, α_{i-1} and β_{i-1} for $i = 1 \sim n - 1$. Considering all possible cases, one has

$$Prob(\tilde{y}_i = 0) = Prob(\tilde{y}_{i-1} = 0)\left(Prob(c_{i-1} = 0)\left(\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2}\right) + Prob(c_{i-1} = 1)\left(\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot 1\right)\right)$$

$$+ Prob(\tilde{y}_{i-1} = 1)\left(Prob(c_{i-1} = 0)\left(\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2}\right) + Prob(c_{i-1} = 1)\left(\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2}\right)\right)$$

$$= \frac{3}{4}Prob(\tilde{y}_{i-1} = 0) + \frac{1}{2}Prob(\tilde{y}_{i-1} = 1)$$

for $i = 1 \sim n - 1$. Solve the iteration function, one can obtain

$$Prob(\tilde{y}_i = 0) = \frac{2}{3} + \frac{1}{3 \cdot 4^i}. \tag{14}$$

From the proof of Property 7, one can first calculate $Prob[c_0] = 1$, $Prob[c_1] = Prob((4\alpha_0 + 2\beta_0 + \tilde{y}_0) \in$

$\{0,6\}) \cdot \frac{1}{2} + Prob((4\alpha_0 + 2\beta_0 + \tilde{y}_0) \in \{2,4\}) \cdot 1 = \frac{1}{4} + \frac{1}{2} = \frac{3}{4}$ and

$$
\begin{aligned}
Prob[c_i] = &\, Prob((4\alpha_i + 2\beta_i) \in \{0,6\})Prob(\tilde{y}_{i-1} = 0)Prob[c_{i-1}]Prob(\text{Condition (11) holds}) \\
&+ Prob((4\alpha_i + 2\beta_i) \in \{0,6\})Prob(\tilde{y}_{i-1} = 1)\left(Prob[c_{i-1}] + (1 - Prob[c_{i-1}])Prob((\alpha_i + x_i) \in \{0,2\})\right) \\
&+ Prob((4\alpha_i + 2\beta_i) \in \{2,4\})Prob(\tilde{y}_{i-1} = 0)Prob[c_{i-1}] + Prob((4\alpha_i + 2\beta_i) \in \{2,4\})Prob(\tilde{y}_{i-1} = 1) \\
= &\, \frac{1}{2}Prob(\tilde{y}_{i-1} = 0)Prob[c_{i-1}]\frac{1}{2} + \frac{1}{2}Prob(\tilde{y}_{i-1} = 1) \cdot \left(Prob[c_{i-1}] + (1 - Prob[c_{i-1}]) \cdot \frac{1}{2}\right) \\
&+ \frac{1}{2}Prob(\tilde{y}_{i-1} = 0)Prob[c_{i-1}] + \frac{1}{2}Prob(\tilde{y}_{i-1} = 1) \\
= &\, Prob[c_{i-1}]\left(\frac{1}{2}Prob(\tilde{y}_{i-1} = 0) + \frac{1}{4}\right) + \frac{3}{4}Prob(\tilde{y}_{i-1} = 1) \\
= &\, Prob[c_{i-1}]\left(\frac{7}{12} + \frac{1}{6 \cdot 4^{i-1}}\right) + \frac{1}{4} - \frac{1}{4^i}
\end{aligned}
\tag{15}
$$

for $i = 2 \sim n-1$, where $Prob[a]$ denotes the probability that the bit a can be confirmed. Finally, one has $Prob[x_0] = \frac{1}{2}$ and

$$
Prob[x_i] = \frac{1}{2}Prob(\tilde{y}_i = 1) + \frac{1}{2}Prob(\tilde{y}_i = 0)Prob[c_i]
\tag{16}
$$

for $i = 1 \sim n-2$. Incorporate Eq. (14) and Eq. (15) into Eq. (16), one can obtain that $Prob[x_0] = \frac{1}{2}$, $Prob[x_1] = 0.4062$, $Prob[x_2] = 0.3818$, and $Prob[x_i] \approx 0.37$ for $i \geq 3$. Now, one can assure that $key1_i$ and $key2_i$ can not be confirmed definitely with a probability smaller than or equal to $(1 - 0.37)^{n_0} = 0.63^{n_0}$ and $0.63^{(8MN/n - n_0)}$ respectively, where n_0 is cardinality of the set $\{k | B(k) \in \{2,3\}, k = 0 \sim 8MN/n - 1\}$. Note that confirmation of the bits of x in Eq. (1) is determined by α, β, and the used unknown variable x, and no any bit can be confirmed under some cases (See the examples shown in Table 2). In addition, as for plaintext chose from natural images, α follows Gaussian distribution. But, we can still believe that the set $\{key1, key2\}$ can be reconstructed in a high probability since the values of n_0 and $8MN/n - n_0$ are both very large in general.

Table 2.   The number of confirmed bits of x in Eq. (1) under some sets of $(\alpha, \beta, \tilde{y})$ when $n = 8$, where $\tilde{y}$ is determined by (α, β, x) via Eq. (1).

x	$(0,1)$	$(17,54)$	$(31,102)$	$(44,93)$	$(51,95)$	$(73,79)$	$(87,122)$	$(125,126)$
7	4	1	3	4	1	0	6	1
8	1	3	1	1	1	0	3	2
9	2	1	3	2	1	1	3	1
28	1	7	1	1	2	1	2	2
59	3	4	6	7	3	1	3	1
71	8	1	3	4	1	0	6	1
76	1	6	1	1	1	1	2	2
95	6	2	5	5	0	0	6	1
99	3	3	3	4	1	2	4	1
111	5	1	4	1	0	0	5	1
125	2	4	7	2	0	1	2	1
127	7	1	6	6	0	0	5	1

Referring to Proposition 1 and Eq. (3), one can obtain the scope of $B(k)$,

$$
\mathbb{B}(k) = \begin{cases} \{1,3\} & \text{if } (J_1'(k) \oplus J_1(k)) \bmod 2 = 0; \\ \{0,2\} & \text{otherwise,} \end{cases}
\tag{17}
$$

for $k = 0 \sim 8MN/n - 1$.

Proposition 1. *Assume that a and x are both n-bit integers and $n \in \mathbb{Z}^+$, $((a \dotplus x) \oplus x)$ has the same parity as a and $((a \dotplus x) \odot x)$ has opposite parity as a.*

Proof.   Existence of four equations

$$((1 + x_0) \bmod 2) \oplus x_0 \equiv 1,$$
$$((0 + x_0) \bmod 2) \oplus x_0 \equiv 0,$$
$$((1 + x_0) \bmod 2) \odot x_0 \equiv 0,$$
$$((0 + x_0) \bmod 2) \odot x_0 \equiv 1,$$

is independent of x_0, so the proposition is proved.   ■

Proposition 2. *Assume that a and x are both n-bit integers, $n \in \mathbb{Z}^+$, one has the following two equations*

$$(a \oplus x) \dot{-} x = (a \oplus x \oplus 2^{n-1}) \dot{-} (x \oplus 2^{n-1}),$$
$$(a \oplus \overline{x}) \dot{-} x = (a \oplus \overline{x \oplus 2^{n-1}}) \dot{-} (x \oplus 2^{n-1}).$$

Proof.   See the proof of Proposition 1 in [Li *et al.*, 2011].   ■

According to the pre-defined condition $key1 \neq key2$, there are only two possible combinations of $key1$ and $key2$. Let $(key1^*, key2^*)$ denote the searched version of $(key1, key2)$. Proposition 2 illustrates that the unknown most significant bit of $key1^* = \sum_{j=0}^{n-1} key1_j^* \cdot 2^j$ and $key2^* = \sum_{j=0}^{n-1} key1_j^* \cdot 2^j$ has no influence on decryption of MCKBA/HCKBA. Then, one can further obtain the approximate value of $B(k)$, $B^*(k)$, and the $n-1$ least significant bits of $key1$ and $key2$ by the following two different ways:

- *W1)* For $k = 0 \sim 8MN/n - 1$, one has

$$B^*(k) = \begin{cases} 3 & \text{if } F(key_1^*, J_1(k), J_1'(k)) = 0, F(key_2^*, J_1(k), J_1'(k)) \neq 0 \\ & \text{or } F(key_1^*, J_2(k), J_2'(k)) = 0, F(key_2^*, J_2(k), J_2'(k)) \neq 0; \\ 2 & \text{if } G(key_1^*, J_1(k), J_1'(k)) = 0, G(key_2^*, J_1(k), J_1'(k)) \neq 0 \\ & \text{or } G(key_1^*, J_2(k), J_2'(k)) = 0, G(key_2^*, J_2(k), J_2'(k)) \neq 0; \\ 1 & \text{if } F(key_1^*, J_1(k), J_1'(k)) \neq 0, F(key_2^*, J_1(k), J_1'(k)) = 0 \\ & \text{or } F(key_1^*, J_2(k), J_2'(k)) \neq 0, F(key_2^*, J_2(k), J_2'(k)) = 0; \\ 0 & \text{if } G(key_1^*, J_1(k), J_1'(k)) \neq, G(key_2^*, J_1(k)^*, J_1'(k)) = 0 \\ & \text{or } G(key_1^*, J_2(k), J_2'(k)) \neq 0, G(key_2^*, J_2(k), J_2'(k)) = 0, \end{cases} \quad (18)$$

where $F(x, \alpha, y) = y - ((\alpha \dot{+} x) \oplus x)$, $G(x, \alpha, y) = y - ((\alpha \dot{+} x) \odot x)$. Note that Eq. (17) makes only two conditions in Eq. (18) need being verified. Obviously, one can assure that $\sum_{j=0}^{n-2} key_{1,j}^* \cdot 2^j = \sum_{j=0}^{n-2} key_{1,j} \cdot 2^j$, $\sum_{j=0}^{n-2} key_{2,j}^* \cdot 2^j = \sum_{j=0}^{n-2} key_{2,j} \cdot 2^j$.

- *W2)* When there exists $i \in \{0, \cdots, n-2\}$ satisfying that $s(k, i) = 1$, one can obtain

$$\mathbb{B}^*(k) = \begin{cases} \{2, 3\} & \text{if } key(k)_i \neq key2_i^*; \\ \{0, 1\} & \text{if } key(k)_i \neq key1_i^*, \end{cases} \quad (19)$$

for $k = 0 \sim 8MN/n - 1$. Then, the value of $B(k)$ can be obtained by setting $B^*(k) = \mathbb{B}^*(k) \bigcap \mathbb{B}(k)$ for $k = 0 \sim 8MN/n - 1$. Finally, one can conclude that $(key1^*, key2^*) = (\sum_{i=0}^{n-2} key1_i^* \cdot 2^i, \sum_{i=0}^{n-2} key2_i^* \cdot 2^i)$, and $\{B^*(k)\}_{k=0}^{8MN/n-1}$ can work together as equivalent secret key of MCKBA/HCKBA due to that $(key1, key2, B(k)) = (a, b, c)$ and $(key1, key2, B(k)) = (b, a, (c+2) \bmod 4)$ are equivalent for Eq. (4).

Now, let's study the success probability of the above two methods. The success of the method *W1)* depends on existence of one of the eight condition in Eq. (18). As $F(x, \alpha, y) = 0$ if and only if $G(x, \alpha, y \oplus (2^n - 1)) = 0$, only one of the two functions need being studed. Obviously,

$$\begin{cases} F(x, 0, y) \equiv y, \\ F(x, 2^{n-1}, y) \equiv y - 2^{n-1}. \end{cases} \quad (20)$$

So, $B(k)$ can not be confirmed by Eq. (18) when $J_1(k) \in \{0, 2^{n-1}\}$ and $J_2(k) \in \{0, 2^{n-1}\}$ exist at the same time. It is very hard to derive the probability for other cases theoretically. Instead, we calculate the probability that $F(x_1, \alpha, y) = F(x_2, \alpha, y) = 0$ via simulation, where

$$\sum_{j=0}^{n-1} (x_{1,j} \oplus x_{2,j}) = \lceil n/2 \rceil, \tag{21}$$

$x_1 = \sum_{j=0}^{n-1} x_{1,j} \cdot 2^j$, $x_2 = \sum_{j=0}^{n-1} x_{2,j} \cdot 2^j$. Assume $key1, key2$ distributes uniformly, the distribution of probability $F(key1, \alpha, y) = F(key2, \alpha, y) = 0$ under different values of α and some values of n is shown in Fig. 1.
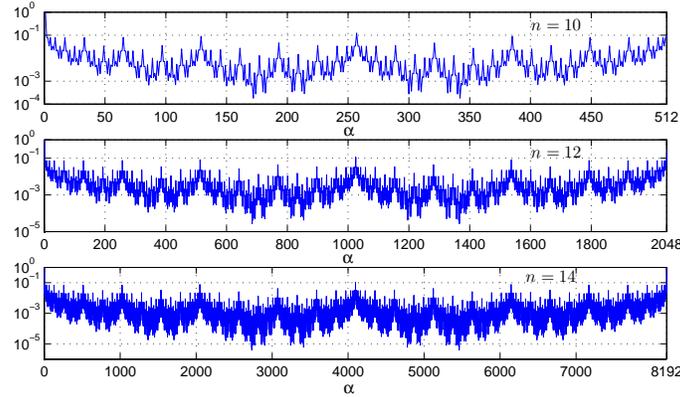

Fig. 1. The probability $F(x_1, \alpha, y) = F(x_2, \alpha, y) = 0$ under different value of α, where x_1, x_2 satisfy the constraint condition (21).

The success of the method *W2) can be analyzed as follows. Assume $key1, key2$ distributes uniformly, one can get the probability that at least one bit (exclude the most significant bit) of $key(k)$ satisfy $s(k, i) = 1$ and one condition of Eq. (19) is*

$$Prob[\text{Eq. (19) holds}] = 1 - \sum_{i=0}^{n-1} \binom{n-1}{i} (Prob[x_i])^i (1 - Prob[x_i])^{n-1-i} \left(\frac{1}{2}\right)^i$$

$$\geq 1 - \sum_{i=0}^{n-1} \binom{n-1}{i} \left(\frac{1}{2}\right)^i \left(1 - \frac{1}{2}\right)^{n-1-i} \left(\frac{1}{2}\right)^i$$

$$= 1 - \left(\frac{1}{2}\right)^{n-1} \sum_{i=0}^{n-1} \binom{n-1}{i} \left(\frac{1}{2}\right)^i$$

$$= 1 - \left(\frac{3}{4}\right)^{n-1}.$$

From the above analysis, one can see that $B(k)$, $k = 0 \sim 8MN/n - 1$, can be determined with a probability larger than $1 - \left(\frac{3}{4}\right)^{n-1}$ when α, β and x in Eq. (1) distributes uniformly. Although pixels of natural images follow Gaussian distribution, and $J_1(k)$ and $J_2(k)$ in Eq. (5) do not distribute uniformly, we still can believe that the success probability of this method is very high since only one bit satisfying the conditions in Eq. (19) is needed, especially when n is relatively large.

To verify the real performance of the above analysis, a number of experiments are carried out on some plain-images of size 512×512 with the method *W1)* when $n = 32$. When $x_0 = 319684607/2^{32}$, $key_1 = 3835288501$, and $key_2 = 1437224678$. Two known plain-images "Peppers" and "Baboon", and the corresponding cipher-images are adopted. Equivalent key $key1^*$, $key2^*$ and $\{B^*(k)\}_{k=0}^{8MN/n-1}$ is used to decrypt another cipher-image shown in Fig. 2a) and the recovered result is shown in Fig. 2b), which is

identical with the original version. From the experiment, we found that only a little pixels (no more than ten pixels for plain-image of size 512×512) are not recovered correctly when $n \leq 28$. This agree with our expectation as the probability that none of condition of Eq. (18) and condition (20) are satisfied become larger when n is smaller.
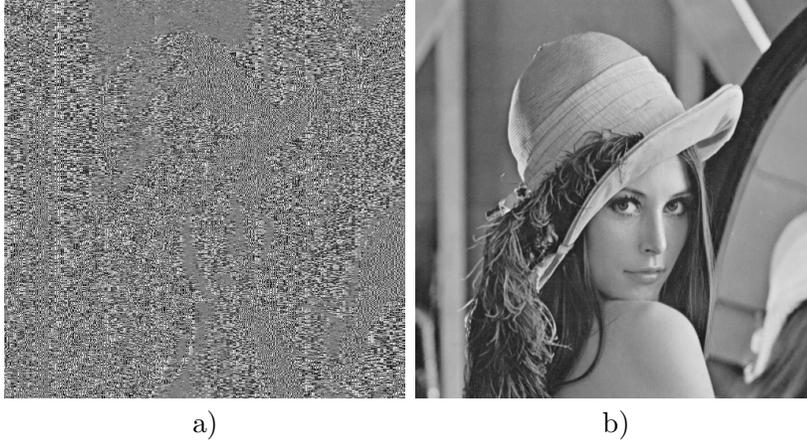

a)   b)

Fig. 2.   The decryption result of another cipher-image encrypted with the same secret key: a) cipher-image; b) decrypted plain-image.

3.3.   *Chosen-plaintext attack*

Chosen-plaintext attack is an enhanced version of known-plaintext attack, where the plaintext can be chosen arbitrarily to obtain the information about the secret key in a more efficient way. In this subsection, the chosen-plaintext attack on MCKBA/HCKBA is briefly introduced due to the following two points: 1) the known-plaintext attack on MCKBA/HCKBA works well in a relatively high probability and the chosen-plaintext version can improve its performance a little; 2) the underlying theorem supporting the attack proposed in [Li *et al.*, 2011, Theorem 1] is not right and corrected in Proposition 3.

Proposition 3.   *Assume that α, β, x are all n-bit integers, then a lower bound on the number of queries (α, β) to solve Eq. (1) in terms of modulo 2^{n-1} for any x is 1 if $n = 2$; 2 if $n > 2$.*

Proof.   When $n = 2$, one can obtain $x_0 = \tilde{y}_1$ by choosing $(\alpha_0, \beta_0) = (1, 0)$. When $n > 2$, $\tilde{y}_1$ may be equal to zero or one no matter what (α_0, β_0) is, which means that it is impossible to satisfy the condition of Property 3 for any x. So, we have to resort to another query (α', β'). Let $\alpha_i', \beta_i', y_i', \tilde{y}_i'$ and c_i' denote the counterparts of $\alpha_i, \beta_i, y_i, \tilde{y}_i$, and c_i corresponding to (α', β'). Given a set of $(\alpha_{i+k}, \beta_{i+k})$ and $(\alpha_{i+k}', \beta_{i+k}')$, one can obtain $(c_{i+k+1}, \tilde{y}_{i+k+1})$ and $(c_{i+k+1}', \tilde{y}_{i+k+1}')$ from $(c_{i+k}, \tilde{y}_{i+k})$ and $(c_{i+k}', \tilde{y}_{i+k}')$, respectively, where i, k are non-negative integers. Let arrows of plain head and "V-back" head denote $x_{i+k} = 0$ and $x_{i+k} = 1$, respectively, Fig. 3 illustrates the mapping relationship between $(c_{i+k}, \tilde{y}_{i+k}, c_{i+k}', \tilde{y}_{i+k}')$ and $(c_{i+k+1}, \tilde{y}_{i+k+1}, c_{i+k+1}', \tilde{y}_{i+k+1}')$ for a given $(\alpha_{i+k}, \beta_{i+k}, \alpha_{i+k}', \beta_{i+k}')$, where $k = 0, 1$. Since $(c_0, \tilde{y}_0, c_0', \tilde{y}_0') \equiv (0, 0, 0, 0)$, the dashed arrows in Fig. 3 describe operations of Eq. (6) in the two least significant bit planes corresponding to two sets of (α, β). Note that the data in the third column is exactly the same as the first one. Therefore, Fig. 3 demonstrates operations of Eq. (6) under all different bit levels if the variable i goes through $2 \cdot t$, where $t = 0 \sim \lfloor n/2 \rfloor$ and $i + k \leq n - 1$. Referring to Fig. 3, it can be easily verified that

$$1 \in \{y_i, y_i'\}$$

is always satisfied, which means that x_i can be derived from Table 1.   ∎

Under scenario of chosen-plaintext attack, one may make the plaintext satisfy that at least one pair of elements in $\{(J_1(k), J_2(k)) \,|\, B(k) \in \{0, 1\}\}$ whose i-th bit plane satisfy the condition of Property 3.
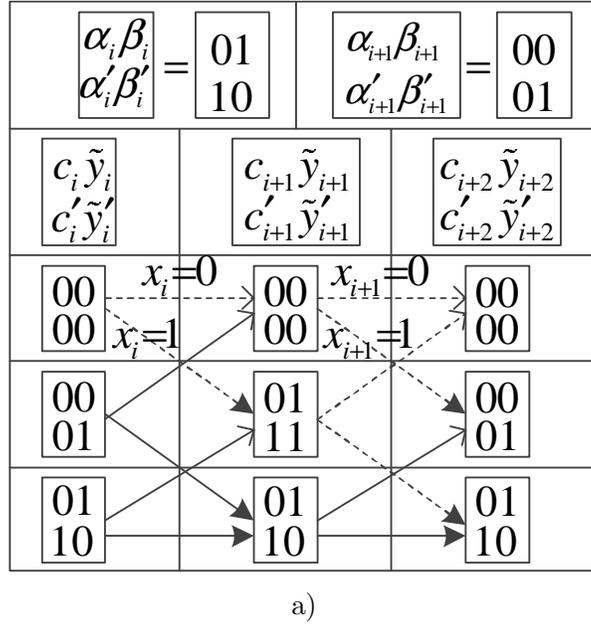
a)

Fig. 3. Relationship between $(c_{i+k}, \tilde{y}_{i+k}, c'_{i+k}, \tilde{y}'_{i+k})$ and $(c_{i+k+1}, \tilde{y}_{i+k+1}, c'_{i+k+1}, \tilde{y}'_{i+k+1})$ for a given $(\alpha_{i+k}, \beta_{i+k}, \alpha'_{i+k}, \beta'_{i+k})$, where $k = 0, 1$.

The same case exists for $\{(J_1(k), J_2(k)) \,|\, B(k) \in \{2, 3\}\}$. The expected chosen-plaintext can be obtained in a high probability by assigning $(J_1(k), J_2(k))$ with one of the two sets of number given in Corollary 3.1 randomly. Compared with the known-plaintext attack, the chosen-plaintext attack has the following two superior performances: 1) the set $\{key1, key2\}$ can be reconstructed with much less complexity and much higher degree of accuracy; 2) the bits of $key(k)$ can be confirmed with a little higher probability.

Corollary 3.1. *The $(n-1)$ least significant bits of x in Eq. (1) can be determined easily by setting (α, β) with the following two sets of numbers*

$$\left\{ \left(\sum_{j=0}^{\lceil n/2 \rceil - 1} (00)_2 \cdot 4^j \right) \bmod 2^n, \left(\sum_{j=0}^{\lceil n/2 \rceil - 1} (10)_2 \cdot 4^j \right) \bmod 2^n \right\},$$

$$\left\{ \left(\sum_{j=0}^{\lceil n/2 \rceil - 1} (10)_2 \cdot 4^j \right) \bmod 2^n, \left(\sum_{j=0}^{\lceil n/2 \rceil - 1} (01)_2 \cdot 4^j \right) \bmod 2^n \right\},$$

and checking the corresponding $\tilde{y} = y \oplus \alpha \oplus \beta$.

Proof.   The proof is straightforward and therefore omitted.   ∎

4. Conclusion

In this paper, the security of the image encryption algorithm MCKBA/HCKBA has been restudied in detail. Based on some properties of a composite function composed of the modulo addition and the XOR operation, a known-plaintext attack and an improved chosen-plaintext attack were provided to determine an equivalent secret key of MCKBA/HCKBA. The cryptanalysis provided in this paper sheds some light on breaking other encryption schemes based on multiple combination of the modulo addition and XOR operations.

Acknowledgement

References

Álvarez, G. & Li, S. [2006] "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos* **16**, 2129–2151.

Chen, F., Wong, K.-W., Liao, X. & Xiang, T. [2012] "Period distribution of generalized discrete arnold cat map for N=pe," *IEEE Transactions on Information Theory* **58**, 445–452.

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

Chen, J., Zhou, J. & Wong, K.-W. [2011] "A modified chaos-based joint compression and encryption scheme," *IEEE Transactions on Circuits and Systems II* **58**, 110–114.

Gangadhar, C. & Rao, K. D. [2010] "Hyperchaos based image encryption," *International Journal of Bifurcation and Chaos* **19**, 3833–3839.

Li, C., Chen, M. Z. Q. & Lo, K.-T. [2011] "Breaking an image encryption algorithm based on chaos," *International Journal of Bifurcation and Chaos* **21**, 3518–3524.

Li, C., Li, S., Asim, M., Nunez, J., Alvarez, G. & Chen, G. [2009] "On the security defects of an image encryption scheme," *Image and Vision Computing* **27**, 1371–1381.

Li, C., Li, S. & Lou, D.-C. [2006] "On the security of the Yen-Guo's domino signal encryption algorithm (DSEA)," *Journal of Systems and Software* **79**, 253–258.

Li, C., Li, S., Zhang, D. & Chen, G. [2005] "Chosen-plaintext cryptanalysis of a clipped-neural-network-based chaotic cipher," *Lecture Notes in Computer Science* **3497**, 630–636.

Li, S., Chen, G. & Mou, X. [2004] "On the security of the Yi-Tan-Siew chaotic cipher," *IEEE Transactions on Circuits and Systems–II: Express Briefs* **51**, 665–669.

Li, S., Li, C., Chen, G. & Lo, K.-T. [2008] "Cryptanalysis of the RCES/RSES image encryption scheme," *Journal of Systems and Software* **81**, 1130–1143.

Li, S. & Zheng, X. [2002] "Cryptanalysis of a chaotic image encryption method," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 708–711.

Paul, S. & Preneel, B. [2005] "Near optimal algorithms for solving differential equations of addition with batch queries," *Lecture Notes in Computer Science* **3797**, 90–103.

Rao, K. & Gangadhar, C. [2007] "Modified chaotic key-based algorithm for image encryption and its VLSI realization," *Proceedings of the 2007 15th International Conference on Digital Signal Processing*, pp. 439–442.

Socek, D., Li, S., Magliveras, S. S. & Furht, B. [2005] "Enhanced 1-D chaotic key-based algorithm for image encryption," *Proceedings of the First IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm 2005)*, pp. 406–408.

Solak, E., Cokal, C., Yildiz, O. T. & Biyikoglu, T. [2010a] "Cryptanalysis of Fridrich's chaotic image encryption," *International Journal of Bifurcation and Chaos* **20**, 1405–1413.

Solak, E., Rhouma, R. & Belghith, S. [2010b] "Cryptanalysis of a multi-chaotic systems based image cryptosystem," *Optics Communications* **283**, 232–236.

Takahashi, Y., Nakano, H. & Saito, T. [2004] "A simple hyperchaos generator based on impulsive switching," *IEEE Transactions on Circuits and Systems II-Express Briefs* **51**, 468–472.

Wang, X., Lai, X., Feng, D., Chen, H. & Yu, X. [2005] "Cryptanalysis of the hash functions MD4 and RIPEMD," *Lecture Notes in Computer Science* **3494**, 1–18.

Xiao, D., Liao, X. & Wong, K.-W. [2006] "Improving the security of a dynamic look-up table based chaotic cryptosystem," *IEEE Transactions on Circuits and Systems II: Express Briefs* **53**, 502–506.

Yen, J.-C. & Guo, J.-I. [2000] "A new chaotic key-based design for image encryption and decryption," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 49–52.

Zhou, J. & Au, O. C. [2011] "On the security of chaotic convolutional coder," *IEEE Transactions on Circuits and Systems I-Regular Papers* **58**, 595–606.