Cryptanalysis of the RCES/RSES Image Encryption Scheme

Shujun Li, Chengqing Li, Guanrong Chen, *Fellow, IEEE* and Xuanqin Mou

***Abstract*— Recently, a chaos-based image encryption scheme called RCES (also called RSES) was proposed. This paper analyzes the security of RCES, and points out that it is insecure against the known/chosen-plaintext attacks: the number of required known/chosen plain-images is only one or two. In addition, the security of RCES against the brute-force attack was overestimated. Both theoretical and experimental analyses are given to show the performance of the suggested known/chosen-plaintext attacks. The insecurity of RCES is due to its special design, which makes it a typical example of insecure image encryption schemes. Some lessons are drawn from RCES to show some common principles for ensuring the high level of security of an image encryption scheme.**

***Index Terms*— image encryption, chaos, chaotic cryptography, RCES/RSES, cryptanalysis, known-plaintext attack, chosen-plaintext attack, CKBA.**

I. Introduction

IN the digital world today, as the rapid progress of multimedia and network technologies, the security of digital images becomes more and more important, since the communications of digital products over networks occur more and more frequently. Furthermore, special and reliable security in storage and transmission of digital images is needed in many applications, such as pay-TV, medical imaging systems, military image database and communications as well as confidential video conferencing, etc. In recent years, some consumer electronic devices, especially mobile phones and hand-held devices, have also started to provide the function of saving and exchanging digital images via the support of multimedia messaging services over wireless networks.

To meet the challenges arising from different applications, good encryption of digital images is necessary. The simplest way to encrypt an image is to consider the 2-D image stream as a 1-D data stream, and then encrypt this 1-D stream with any available cipher [1]. Although such a simple way is sufficient to protect digital images in some civil applications, encryption schemes considering special features of digital images, such as the bulky size and the large redundancy in uncompressed images, are still needed to provide better overall performance and make the adoption of the encryption scheme easier in the whole image processing system. For a more thorough discussion on the problems of the naive use of text ciphers for digital images, see Sec. 4.2.2 of [2].

Since the 1990s, many specific algorithms have been proposed, aiming to provide better solutions to image encryption [3]–[18]. At the same time, cryptanalytic work on proposed image encryption schemes has also been developed, and some existing schemes have been found to be insecure from the cryptographical point of view [19]–[26]. Due to the tight relationship between chaos and cryptography [25, Chap. 2], chaotic systems have been widely used in image encryption to realize diffusion and confusion in a good cipher [10]–[18]. For a comprehensive survey of the state of the art about image encryption schemes, see Sec. 4.3 and Sec. 4.4 of [2].

The present paper focuses on a new chaos-based image encryption scheme proposed by Chen and Yen in [17], [18], which was originally called RSES (random seed encryption system) in [17] and then renamed to be RCES (random control encryption system) in [18]. RCES is actually an enhanced version of a previously-proposed image encryption scheme called

Shujun Li and Guanrong Chen are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon Toon, Hong Kong SAR, P. R. China. E-mails: hooklee@mail.com, gchen@ee.cityu.edu.hk.

Chengqing Li is with the Institute of Image and Computer Graphics, Department of Mathematics, Zhejiang University, Hangzhou 310027, P.R. China. E-mail: swiftsheep@hotmail.com.

Xuanqin Mou is with the Institute of Image Processing, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P. R. China. E-mails: xqmou@xjtu.edu.cn.

CKBA (chaotic key-based algorithm) [16], which has been cryptanalyzed in [24]. The present paper evaluates the security of RCES, and points out that RCES is as weak as CKBA: only one or two known/chosen plain-images are enough to break it. In addition, it will be shown that the security of RCES against brute-force attack was much overestimated.

Due to the special design of RCES, some essential security defects existing in it are very useful for revealing several design principles in a secure image encryption scheme. This magnifies the significance of RCES and also its negative aspects in the cryptanalysis presented below.

This paper is organized as follows. For convenience, some preliminary knowledge of cryptanalytic techniques is firstly given in Sec. II. Section III briefly introduces RCES and its parent version CKBA. A detailed cryptanalysis of RCES is presented in Sec. IV, where some experimental results are given to support the theoretical analysis. Section V discusses some design principles drawn from the essential security defects of RCES. The last section concludes the paper.

II. Preliminaries of Cryptanalysis

To facilitate the following discussion, this section gives a brief introduction to the basic theory of modern cryptology [27]. Cryptology, the technology of encryption, is composed of two parts: cryptography and cryptanalysis. The former studies how to design good encryption algorithms, and the latter tries to find security weaknesses of proposed algorithms and studies whether or not they are vulnerable to some attacks.

An encryption system is also called a *cipher*, or a *cryptosystem*. The message for encryption is called *plaintext*, and the encrypted message is called *ciphertext*. Denote the plaintext and the ciphertext by P and C, respectively. The encryption procedure of a cipher can be described as $C = E_{K_e}(P)$, where K_e is the encryption key and $E(\cdot)$ is the encryption function. Similarly, the decryption procedure is $P = D_{K_d}(C)$, where K_d is the decryption key and $D(\cdot)$ is the decryption function. When $K_e = K_d$, the cipher is called a *private-key* cipher or a *symmetric* cipher. For private-key ciphers, the encryption-decryption key must be transmitted from the sender to the receiver via a separate secret channel. When $K_e \neq K_d$, the cipher is called a *public-key* cipher or a *asymmetric* cipher. For public-key ciphers, the encryption key K_e is published, and the decryption key K_d is kept private, for which no additional secret channel is needed for key transfer.

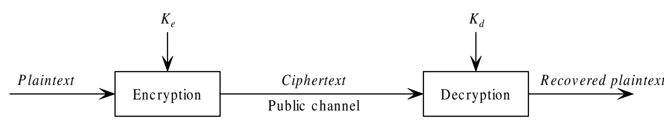


Fig. 1. The encryption and decryption of a cipher

Following Kerckhoff's principle widely acknowledged in the cryptology community [27], the security of a cipher only relies on the decryption key K_d, and it is assumed that all details of the encryption/decryption procedure are known to attackers. Thus, the main task of cryptanalysis is to reconstruct the key, or its equivalent form that can successfully decrypt all or part of the plaintexts.

A cryptographically strong cipher should be secure enough against all kinds of attacks. For most ciphers, the following four attacks under different scenarios should be checked:

- *the ciphertext-only attack* - attackers can only observe some ciphertexts;
- *the known-plaintext attack* - attackers can get some plaintexts and the corresponding ciphertexts;
- *the chosen-plaintext attack* - attackers can choose some plaintexts and get the corresponding ciphertexts;
- *the chosen-ciphertext attack* - attackers can choose some ciphertexts and get the corresponding plaintexts.

The last two attacks, which seem to seldom occur in practice, are feasible in some real applications [27, Sec. 1.1.7] and they become more and more common in the digital world today.

As surveyed in [2], it is known that many image/video encryption schemes are not secure enough against known/chosen-plaintext attacks. This paper shows that RCES is also insecure against known/chosen-plaintext attacks.

III. Introduction to RCES

A. CKBA [16] - The Parent Version of RCES

Assume that the size of the plain-image for encryption is $M \times N$, CKBA can be described as follows.

1) The secret key: two bytes $key1$, $key2$, and the initial condition $x(0) \in (0,1)$ of the following chaotic Logistic map:

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

which is a well-studied chaotic system in chaos theory and behaves chaotically when $\mu > 3.5699\cdots$ [28].

2) Initialization: run the chaotic system to generate a chaotic sequence, $\{x(i)\}_{i=0}^{\lceil MN/8 \rceil - 1}$, where $\lceil a \rceil$ denotes the smallest integer that is not less than a. From the 16-bit binary representation of $x(i) = 0.b(16i+0)b(16i+1)\cdots b(16i+15)$, derive a pseudo-random binary sequence (PRBS), $\{b(i)\}_{i=0}^{2MN-1}$.

3) Encryption: for the plain-pixel $f(x,y)$ $(0 \le x \le M-1, 0 \le y \le N-1)$, the corresponding cipher-pixel $f'(x,y)$ is determined by the following rule:

$$f'(x,y) = \begin{cases} f(x,y) \oplus key1, & B(x,y) = 3, \\ f(x,y) \odot key1, & B(x,y) = 2, \\ f(x,y) \oplus key2, & B(x,y) = 1, \\ f(x,y) \odot key2, & B(x,y) = 0, \end{cases} \tag{2}$$

where $B(x,y) = 2 \times b(x \times N + y) + b(x \times N + y + 1)$, and $\oplus$ and $\odot$ denote XOR and XNOR operations, respectively. Since $a \odot b = \overline{a \oplus b} = a \oplus \bar{b}$, the above equation is equivalent to

$$f'(x,y) = \begin{cases} f(x,y) \oplus key1, & B(x,y) = 3, \\ f(x,y) \oplus \overline{key1}, & B(x,y) = 2, \\ f(x,y) \oplus key2, & B(x,y) = 1, \\ f(x,y) \oplus \overline{key2}, & B(x,y) = 0. \end{cases} \tag{3}$$

4) Decryption: the decryption procedure is like that of the encryption, since $\oplus$ is an involutive operation.

5) A constraint: because not all values of $key1$ and $key2$ can make well-disorderly cipher-images, it is required that $key1$ and $key2$ have 4 different bits (a half of all). In fact, this constraint ensures that the encryption results of $key1$ and $key2$ are sufficiently far.

In [24], CKBA was cryptanalyzed and the following facts were pointed out:

- the security of CKBA against the brute-force attack was over-estimated;
- CKBA is not secure against known/chosen-plaintext attacks, since only one known/chosen plain-image is enough to get an equivalent key, a mask image f_m, by XORing the plain-image f and the cipher-image f', pixel by pixel: $f_m = f \oplus f'$;
- it is easy to reconstruct the whole secret key $\{key1, key2, x(0)\}$ from the mask image f_m, for which the required complexity is rather small.

Apparently, the insecurity of CKBA against known/chosen-plaintext attacks is determined by the fact that $f(x,y) \oplus f'(x,y)$ is fixed to be one of the four values, $key1$, $\overline{key1}$, $key2$, $\overline{key2}$, at any given position (x,y). In fact, for any plain-images, f_1, f_2 and their cipher-images, f_1', f_2', one has

$$f_1(x,y) \oplus f_1'(x,y) = f_2(x,y) \oplus f_2'(x,y) \equiv f_m(x,y)$$

for any position (x,y). As a result, given any cipher-image f', the plain-image can be decrypted as follows: $f = f' \oplus f_m$.

B. RCES [18] (or RSES [17])

RCES is an enhanced version of CKBA, by making $key1$ and $key2$ time-variant, and by introducing a simple permutation operation, $Swap_b(x_1, x_2)$, which exchanges the values of x_1 and x_2 if $b = 1$ and does nothing if $b = 0$.

RCES encrypts plain-images block by block, where each block contains 16 consecutive pixels. To simplify the following description, without loss of generality, assume that the sizes of plain-images are all $M \times N$ (M is the width and N is the height), and that MN can be divided by 16. Consider a plain-image $\{f(x,y)\}_{x=0,y=0}^{x=M-1,y=N-1}$ as a 1-D pixel-sequence $\{f(l)\}_{l=0}^{MN-1}$ by scanning it line by line from bottom to top. The plain-image can be divided into $MN/16$ blocks:

$$\{f^{(16)}(0), \cdots, f^{(16)}(k), \cdots, f^{(16)}(MN/16-1)\},$$

where

$$f^{(16)}(k) = \{f(16k+0), \cdots, f(16k+i), \cdots, f(16k+15)\}.$$

For the k-th pixel-block $f^{(16)}(k)$, the work mechanism of RCES can be described as follows.

1) The secret key: the control parameter μ and the initial condition $x(0)$ of the Logistic map (1).

2) Initialization: run the Logistic map to generate a chaotic sequence, $\{x(i)\}_{i=0}^{MN/16-1}$, and then extract the 24-bit representation of $x(i)$ to yield a PRBS $\{b(i)\}_{i=0}^{3MN/2-1}$. Note that the Logistic map is realized in 24-bit fixed-point arithmetic.

3) Encryption: two pseudo-random seeds,

$$Seed1(k) = \sum_{i=0}^{7} b(24k+i) \times 2^{7-i}, \tag{4}$$

$$Seed2(k) = \sum_{i=0}^{7} b(24k+8+i) \times 2^{7-i}, \tag{5}$$

are calculated to encrypt the current plain-block with the following two steps:

a) Pseudo-randomly swapping adjacent pixels: for $i = 0 \sim 7$, do

$$Swap_{b(24k+16+i)}(f(16k+2i), f(16k+2i+1)). \tag{6}$$

b) Masking the current plain-block with the two pseudo-random seeds: for $j = 0 \sim 15$, do

$$f'(16k+j) = f(16k+j) \oplus Seed(16k+j), \tag{7}$$

where

$$Seed(16k+j) = \begin{cases} Seed1(k), & B(k,j)=3, \\ \overline{Seed1(k)}, & B(k,j)=2, \\ Seed2(k), & B(k,j)=1, \\ \overline{Seed2(k)}, & B(k,j)=0, \end{cases} \tag{8}$$

and $B(k,j) = 2 \times b(24k+j) + b(24k+j+1)$.

4) Decryption: The decryption procedure is similar to the encryption procedure, but the masking operation is exerted before the swapping for each pixel-block.

IV. Cryptanalysis of RCES

Although RCES is much more complicated than CKBA, as analyzed below, its security is not really enhanced by the introduced design complexity.

In this section, the following results are obtained on the security of RCES: 1) its security against brute-force attack was over-estimated; 2) it is not secure against known/chosen-plaintext attacks, and the number of required plain-images is only $O(1)$ and, in fact, only one or two; 3) there are two available known/chosen-plaintext attacks, and they can be further combined to make a nearly-perfect attack to RCES; 4) the chosen-plaintext attacks can even achieve much better breaking performance than their known-plaintext versions.

A. The Brute-Force (Ciphertext-Only) Attack

In [17], [18], Chen and Yen claimed that the complexity of RCES against brute-force attack is $O\left(2^{3MN/2}\right)$ since $\{b(i)\}_{i=0}^{3MN/2-1}$ has $3MN/2$ bits. However, such a statement is not true due to the following reason: all $3MN/2$ bits are uniquely determined by the control parameter μ and the initial condition $x(0)$ of the Logistic map (1), which has only 48 secret bits. This means that the key entropy of RCES is only 48. Considering not all values of μ can produce chaoticity in the Logistic map, the key entropy should be even smaller than 48. To simplify the following analysis, assume that the key entropy is $K_\mu < 48$, so the total number of all possible keys for brute-force search is only 2^{K_μ}.

Considering that the complexity of RCES is $O(MN)$ [18, Sec. 2.4], the complexity against the brute-force attack is $O\left(2^{K_\mu} \cdot MN\right)$. Assume $K_\mu = 48$, for a typical image whose size is 256×256, the complexity is about $O\left(2^{64}\right)$, which is much smaller than $O\left(2^{3MN/2}\right) = O\left(2^{98304}\right)$, the claimed complexity in [17], [18]. Apparently, the security of RCES against the brute-force attack was over-estimated by too much.

B. Known-Plaintext Attack 1: Breaking RCES with a Mask Image f_m

Although different seeds are used for pixels at different positions and pseudo-random swapping operations are exerted on the plain-image before masking, the known-plaintext attack breaking CKBA can be efficiently extended to break RCES. With only one known plain-image and its corresponding cipher-image, it is very easy to get a mask image f_m, which can be used as an equivalent key of the secret key $(\mu, x(0))$ to decrypt any cipher-image whose size is not larger than the size of f_m. When two or more plain-images are known, a swapping matrix Q can be constructed to enhance the breaking performance of the mask image f_m.

1) Get f_m from One Known Plain-Image: Assume that an $M \times N$ plain-image f_K and its corresponding cipher-image f'_K have been known to an attacker. Similar to the way to get the mask image in the known-plaintext attack to CKBA, the attacker here can get f_m by simply XORing the plain-image and the cipher-image pixel by pixel: $f_m(l) = f_K(l) \oplus f'_K(l)$, where $l = 0 \sim MN - 1$.

With the mask image f_m, the attacker tries to recover the plain-image by XORing the mask image and the cipher-image pixel by pixel: $f(l) = f'(l) \oplus f_m(l)$. If a pixel $f(l)$ is not swapped, $f(l) = f'(l) \oplus f_m(l)$; otherwise, $f(l) = f'(l) \oplus f_m(l)$ is generally not true. Assume that the bit $b(24k + 16 + i)$ in Eq. (6) satisfies the balanced distribution[1] over $\{0, 1\}$, it is expected that about half of all plain-pixels are not swapped and can be successfully decrypted with $f_m \oplus f'$. Intuitively, half of plain-pixels should be enough to reveal the main content and some details of the plain-image.

With the secret key $(\mu, x(0)) = (3.915264, 0.2526438)$, which is randomly chosen with the standard `rand()` function, some experiments are made to show the real performance of the mask image f_m in this attack. One known plain-image f_{Lenna} and its cipher-image f'_{Lenna} are shown in Fig. 2. The mask image $f_m = f_{\text{Lenna}} \oplus f'_{\text{Lenna}}$ is given in Fig. 3. For an unknown plain-image f_{Peppers} (Fig. 4a), the mask image f_m is used to recover it from its cipher-image f'_{Peppers} (Fig. 4b). The recovered plain-image $f^*_{\text{Lenna}} = f_m \oplus f'_{\text{Peppers}}$ and the difference image $\left| f^*_{\text{Peppers}} - f_{\text{Peppers}} \right|$ are shown in Fig. 5a and 5b, respectively. It is surprisingly seen that the decryption performance is much better than expected: most (much more than 50%) pixels are successfully recovered, and almost all subtle details remain.

Although the difference $\left| f^*_{\text{Peppers}} - f_{\text{Peppers}} \right|$ visually

[1]Strictly speaking, the Logistic map cannot guarantee the balance of each generated bit, since its variant density function is not uniform [29]. In this paper, without loss of generality, it is taken for granted so as to simplify the theoretical analyses.

a) The known plain-image f_{Lenna} b) The cipher-image f'_{Lenna}

Fig. 2. One known plain-image, 'Lenna' (256×256), and its cipher-image

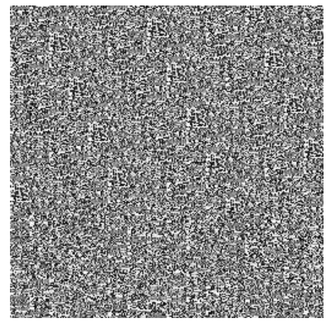
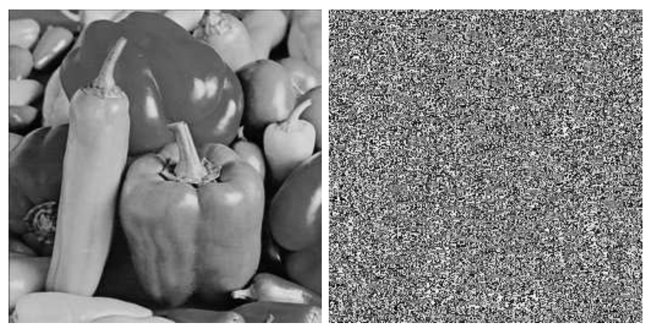
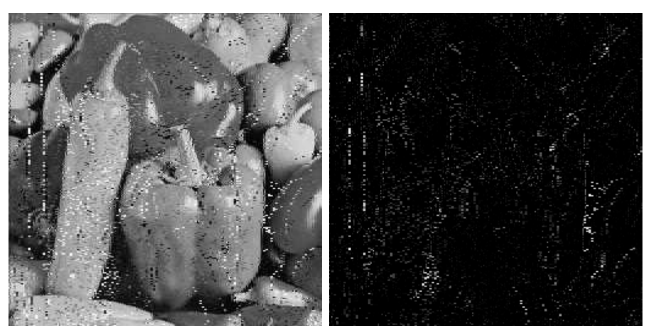


Fig. 3. The mask image f_m derived from f_{Lenna} and f'_{Lenna}



a) The plain-image f_{Peppers} b) The cipher-image f'_{Peppers}

Fig. 4. A plain-image unknown to the attacker, 'Peppers' (256×256), and its cipher-image



a) The recovered plain-image f^*_{Peppers} with f_m b) The recovery difference $\left| f^*_{\text{Peppers}} - f_{\text{Peppers}} \right|$

Fig. 5. Breaking the plain-image with f_m derived from Lenna.bmp

shows that most plain-pixels are exactly recovered, statistical data show that 33,834 pixels in $f^*_{\text{Peppers}} - f_{\text{Peppers}}$ are not zero, i.e., about 51.63% of pixels are not exactly recovered. To explain why f_m is so effective to recover most pixels of the plain-image with only half exactly-recovered pixels, consider two pixels in the known plain-image, $f(2i)$, $f(2i+1)$, and their cipher-pixels, $f'(2i)$, $f'(2i+1)$, where $i = 0 \sim MN/2-1$. Then, the corresponding elements of the two pixels in the mask image f_m will be $f_m(2i) = f(2i) \oplus f'(2i)$ and $f_m(2i+1) = f(2i+1) \oplus f'(2i+1)$. Since all recovery errors are introduced at the positions where the adjacent plain-pixels are swapped, one can theoretically study the recovery performance of the mask image f_m by considering the elements corresponding to the swapped pixels only. Assume that $f(2i)$ and $f(2i+1)$ are swapped in the encryption procedure, $f'(2i) = f(2i+1) \oplus Seed(2i)$ and $f'(2i+1) = f(2i) \oplus Seed(2i+1)$. Therefore,

$$
\begin{aligned}
f_m(2i) &= f^{(\oplus)}(2i) \oplus Seed(2i), && (9)\\
f_m(2i+1) &= f^{(\oplus)}(2i) \oplus Seed(2i+1), && (10)
\end{aligned}
$$

where $f^{(\oplus)}(2i) = f(2i) \oplus f(2i+1)$.

Consider a cipher-image f_1' and its corresponding plain-image f_1. Assuming that the plain-image recovered from f_m is f_1^*, the recovered plain-pixels, $f_1^*(2i)$ and $f_1^*(2i+1)$, satisfy the following propositions and corollaries, respectively.

Proposition 1: $f_1^*(2i) \oplus f_1(2i) = f_1^*(2i+1) \oplus f_1(2i+1) = f^{(\oplus)}(2i) \oplus f_1^{(\oplus)}(2i)$.

Proof: From Eq. (9) and $f_1'(2i) = f_1(2i+1) \oplus Seed(2i)$,

$$
\begin{aligned}
f_1^*(2i) &= f_m(2i) \oplus f_1'(2i),\\
&= \left(f^{(\oplus)}(2i) \oplus Seed(2i)\right)\\
&\quad \oplus \left(f_1(2i+1) \oplus Seed(2i)\right)\\
&= f^{(\oplus)}(2i) \oplus f_1(2i+1)
\end{aligned}
$$

Then, one has

$$
\begin{aligned}
f_1^*(2i) \oplus f_1(2i) &= f^{(\oplus)}(2i) \oplus f_1(2i+1) \oplus f_1(2i)\\
&= f^{(\oplus)}(2i) \oplus f_1^{(\oplus)}(2i).
\end{aligned}
$$

In a similar way, one can get $f_1^*(2i+1) \oplus f_1(2i+1) = f^{(\oplus)}(2i) \oplus f_1^{(\oplus)}(2i)$. Thus, the proof is completed. ■

Proof: This proposition can be proved in a way similar to the proof of the above proposition. ■

Corollary 1: When $f(2i) = f(2i+1)$, $f_1^*(2i) = f_1(2i+1)$ and $f_1^*(2i+1) = f_1(2i)$.

Proof: The results of this corollary are special cases of the above two propositions with $f^{(\oplus)}(2i) = 0$. ■

Based on the above propositions, one can get an upper bound of the recovery errors $|f_1^*(2i) - f_1(2i)|$ and $|f_1^*(2i+1) - f_1(2i+1)|$. Firstly, a lemma should be introduced.

Lemma 1: If $a \oplus b = c$, then $|a - b| \leq c$.

Proof: Represent c in the following binary form:

$$c = (0, \cdots, 0, c_{n-1} = 1, \cdots, c_i, \cdots, c_1, c_0)_2.$$

Similarly, represent a and b as follows:

$$
\begin{aligned}
a &= (a_{N-1}, \cdots, a_{n-1}, \cdots, a_i, \cdots, a_1, a_0)_2,\\
b &= (b_{N-1}, \cdots, b_{n-1}, \cdots, b_i, \cdots, b_1, b_0)_2.
\end{aligned}
$$

From $a \oplus b = c$, one have $\forall j = n \sim N-1$, $a_j = b_j$. Therefore,

$$|a - b| = \sum_{i=0}^{N-1} (a_i - b_i) \cdot 2^i = \sum_{i=0}^{n-1} (a_i - b_i) \cdot 2^i.$$

Since $a_i - b_i \leq |a_i - b_i| = a_i \oplus b_i = c_i$, one has $|a - b| \leq \sum_{i=0}^{n-1} c_i \cdot 2^i = c$. The lemma is thus proved. ■

Corollary 2: $|f_1^*(2i) - f_1(2i)| \leq f^{(\oplus)}(2i) \oplus f_1^{(\oplus)}(2i)$, and $|f_1^*(2i+1) - f_1(2i+1)| \leq f^{(\oplus)}(2i) \oplus f_1^{(\oplus)}(2i)$.

Proof: This corollary is an obvious result of Proposition 1 and Lemma 1. ■

Corollary 2 says that the recovery errors of both $f_1^*(2i)$ and $f_1^*(2i+1)$ will not be larger than $f^{(\oplus)}(2i) \oplus f_1^{(\oplus)}(2i) = f(2i) \oplus f(2i+1) \oplus f_1(2i) \oplus f_1(2i+1)$. Due to the strong correlation between adjacent pixels of digital images, the distribution of the difference between two adjacent pixels is Gaussian-like. As a result, $f^{(\oplus)}(2i)$ will also obeys a (positive) single-side Gaussian-like distribution, which means that the recovery error of each plain-pixel recovered from f_m will also obey a Gaussian-like distribution. The Gaussian-like distribution of recovery errors actually implies that most recovered pixels are close to the real values of the original plain-pixels. Therefore, the surprising recovery performance of f_m shown in Fig. 5 can be naturally explained.

For the plain-image f_{Peppers}, the histograms of some differential images are plotted to verify the above-mentioned theoretical results. Define two $(M-1)\times N$ differential images $f^{(-)}$ and $f^{(\oplus)}$:

$$f^{(-)}(x,y) = f(x,y) - f(x+1,y), \quad (11)$$

$$f^{(\oplus)}(x,y) = f(x,y) \oplus f(x+1,y), \quad (12)$$

where $x = 0 \sim M-2$, $y = 0 \sim N$. The histograms of the above two differential images of f_{Peppers} are shown in Fig. 6. When $f = f_{\text{Lenna}}$, $f_1 = f_{\text{Peppers}}$, the histograms of $f^{(\oplus)} \oplus f_1^{(\oplus)}$ and $\left|f^*_{\text{Peppers}} - f_{\text{Peppers}}\right|$ are shown in Fig. 7. Apparently, Figure 7 agrees with Corollary 2 very well. Note that only the swapped pixels are enumerated for the histogram of $\left|f^*_{\text{Peppers}} - f_{\text{Peppers}}\right|$, since the above theoretical analysis on the recovery errors is only focused on the swapped pixels.

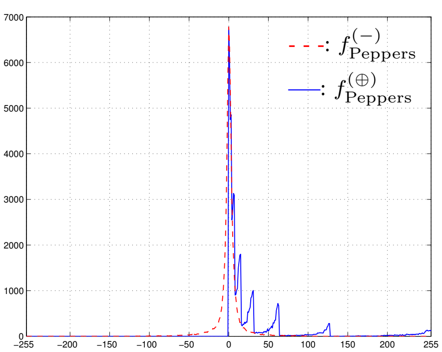


Fig. 6. The histograms of $f^{(-)}_{\text{Peppers}}$ and $f^{(\oplus)}_{\text{Peppers}}$

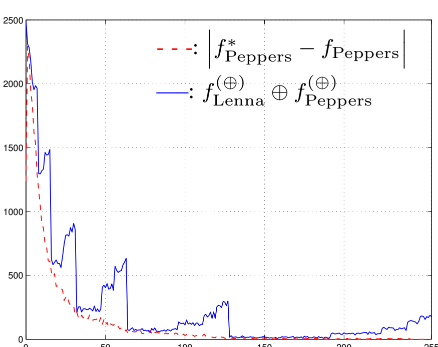


Fig. 7. The histograms of $f^{(\oplus)}_{\text{Lenna}} \oplus f^{(\oplus)}_{\text{Peppers}}$ and $\left|f^*_{\text{Peppers}} - f_{\text{Peppers}}\right|$

Since all recovery errors are introduced by swapped pixels, the recovery performance will be better if some swapped pixels can be distinguished. In the following, it is shown that an attacker can manage to do so by manually detecting visible noises in cipher-images, and by intersecting multiple mask images generated from different known plain-images.

2) Amending f_m with More Cipher-Images: Assume that the corresponding plain-image of a cipher-image does not contain salt-pepper impulsive noises. Then, one can assert that all such noises in the recovered plain-image indicates the positions of swapped pixels. Observing the recovered plain-image f^*_{Lenna} shown in Fig. 5a, one can find many distinguishable noises by naked eyes, which correspond to the strong edges of the known plain-image f_{Lenna} (see Fig. 5b). Following Proposition 1, strong edges means large values of $f^{(\oplus)}(x)$, and so generates salt-pepper noises.

Once some swapped pixels are distinguished, one can generate a swapping $(0,1)$-matrix $Q = [q_{i,j}]_{M\times N}$, where $q_{i,j} = 1$ for swapped pixels and $q_{i,j} = 0$ for others. Similarly, Q can be represented in 1-D form: $Q = \{q(l)\}_{i=0}^{MN-1}$. With the swapping matrix, the mask image f_m is amended as follows: for $i = 0 \sim MN/2-1$, if $q(2i) = 1$ or $q(2i+1) = 1$, the values of $f_m(2i)$ and $f_m(2i+1)$ are re-calculated as follows: $f_m(2i) = f(2i) \oplus f'(2i+1)$ and $f_m(2i+1) = f(2i+1) \oplus f'(2i)$; otherwise, $f_m(2i)$ and $f_m(2i+1)$ are left untouched. With the amended f_m and the swapping matrix Q, one can decrypt the cipher-images in the following two steps:

- use f_m to XOR the cipher-image to get an initial recovered plain-image f^*;
- $\forall i = 0 \sim MN/2-1$, if $q(2i) = 1$ or $q(2i+1) = 1$, swap the two adjacent pixels $f^*(2i)$ and $f^*(2i+1)$.

If an attacker can get more cipher-images encrypted with the same key, he can distinguish more swapped pixels, and gets better recovery performance with f_m and Q. This implies that more and more knowledge on how to purify the attack can be learned from the cipher-images, which is a very interesting and useful feature from an attacker's point of view.

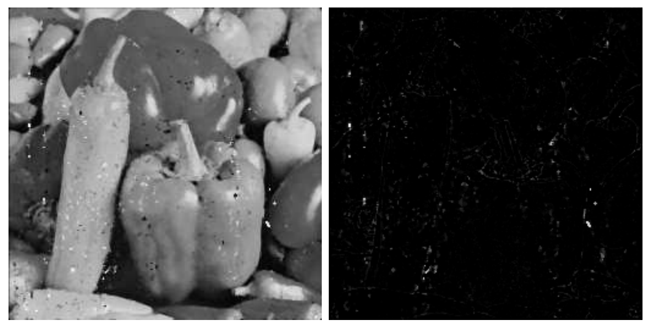
3) Amending f_m with More Known Plain-Images: With two or more known plain-images and their cipher-images encrypted with the same secret key, it is possible to successfully distinguish most swapped pixels, achieving nearly perfect recovery performance. Given $n \geq 2$ known plain-images, $f_1, \cdots, f_n$, and their cipher-images, $f'_1, \cdots, f'_n$, one can get n mask images $f_m^{(i)} = f_i \oplus f_i'$ ($i = 1 \sim n$). Apparently, if the l-th pixel is not swapped, $\forall i \neq j$, $f_m^{(i)}(l) = f_m^{(j)}(l)$. That is, if $f_m^{(i)}(l) \neq f_m^{(j)}(l)$, it can be asserted that the pixel at this position is swapped. Therefore, by comparing the elements of n mask images, some positions corresponding to the swapped pixels can be distinguished. With the swapping information, following the same way described above, a swapping matrix Q can be constructed, and then f_m is amended with Q with the way mentioned above. Using the amended f_m and the swapping matrix Q, the cipher-image is decrypted with XOR and swapping operations.

From Eqs. (9) and (10), the probability of $f_m^{(i)}(l) \neq f_m^{(j)}(l)$ is the probability of $f_i^{(\oplus)}(2i) \neq f_j^{(\oplus)}(2i)$, where $l = 2i$ or $2i+1$. Assume the n mask images are independent of each other and the value of each element distributes uniformly over $\{0, \cdots, 255\}$. The probability of $f_m^{(i)}(l) \neq f_m^{(j)}(l)$ will be $1 - 256^{-1} \approx 0.996$. This means that only two mask images are enough to distinguish almost all swapped pixels. However, since the mask images are generally not independent of each other and $f_m(l)$ does not obey uniform distribution, the real probability will be less than $1 - 256^{-1}$. Fortunately, for most natural images, this probability is still sufficiently close to $1 - 256^{-1}$, so that two known plain-images are still enough to distinguish most swapped pixels. Given two known plain-images, 'Lenna' (see Fig. 2a) and 'Babarra' (see Fig. 8a), the recovery performance of the attack to 'Peppers' is shown in Fig. 8b. It can be seen that the recovered plain-image is almost perfect, and only 952 (about 1.45% of all) pixels are not exactly recovered.

4) Enhancing the Recovered Plain-Image with Image Processing Techniques: To further improve the visual quality of the recovered plain-images, some noise reducing techniques can be used to further reduce the recovery errors. For the recovered plain-image f_{Lenna}^* in Fig. 5a, the enhanced plain-image f_{Lenna}^* with a 3×3 median filter and the corresponding difference image $\left|f_{\text{Peppers}}^* - f_{\text{Peppers}}\right|$ are shown in Fig. 9a and 9b, respectively. It can be seen that the visual quality of f_{Lenna}^* is enhanced significantly. Note that more complicated image processing techniques are still available to further polish the recovered plain-image, one of which will be introduced below in Sec. IV-E.

a) Another known plain-image f_{Babarra} b) The recovered plain-image f_{Peppers}^{**}

Fig. 8. The recovery performance on 'Peppers' with two known plain-images: 'Lenna' and 'Babarra' (both 256×256)



a) The plain-image f_{Peppers}^* enhanced with a 3×3 median filter b) The difference image $\left|f_{\text{Peppers}}^* - f_{\text{Peppers}}\right|$

Fig. 9. Enhancing the recovered plain-image with a 3×3 median filter

C. Known-Plaintext Attack 2: Breaking the Chaotic Map

In the above-discussed attack based on mask images, assuming that the size of f_m is $M \times N$, it is obvious that only $M\times N$ leading pixels in a larger cipher-image can be recovered with f_m (and perhaps Q). To decrypt more pixels, the secret control parameter μ and a chaotic state $x(k)$ occurring before $x(MN/16-1)$ have to be known, so that one can calculate more chaotic states after $x(MN/16-1)$. That is, the chaotic map should be found. Actually, it is possible for an attacker to achieve this goal with a high probability and a sufficiently small complexity, even when only one plain-image is known. Similarly, the more the number of known plain-images are, the closer the probability will be to 1, the smaller the value of k will be, and the lower the attack complexity will be.

1) Guessing a Chaotic State $x(k)$ from f_m: In the k-th pixel-block, for any unswapped pixel $f(16k+j)$,

$$f_m(16k+j) = f(16k+j) \oplus f'(16k+j) = Seed(16k+j),$$

which must be one value in the set

$$S_4 = \left\{ Seed1(k), \overline{Seed1(k)}, Seed2(k), \overline{Seed2(k)} \right\}. \quad (13)$$

Therefore, if there are enough unswapped pixels, the right values of $Seed1(k)$ and $Seed2(k)$ can be guessed by enumerating all 2-value and 1-value[2] combinations of $f_m(16k+0) \sim f_m(16k+15)$. To eliminate most wrong values of $Seed1(k), Seed2(k)$, the following requirements are useful:

- both $B(k,j)$ and $(Seed1(k), Seed2(k))$ are generated with $\{b(24k+j\}_{j=0}^{15}$;
- $Seed(16k+j)$ is uniquely determined by $B(k,j)$ and $Seed1(k), Seed2(k)$ following Eq. (8).

For each guessed values passing the above requirements, the corresponding chaotic state $x(k) = 0.b(24k+0)\cdots b(24k+23)$ is derived as follows:

- reconstruct $\{b(24k+i)\}_{i=0}^{15}$ from $Seed1(k)$, $Seed2(k)$;
- reconstruct $\{b(24k+16+i)\}_{i=0}^{7}$ with the following rule: if both $f_m(16k+2i) \in S_4$ and $f_m(16k+2i+1) \in S_4$ hold, $b(24k+16+i)=1$, else $b(24k+16+i)=0$.

Note that some extra errors will be introduced in the least 8 bits $\{b(24k+16+i)\}_{i=0}^{7}$, which makes the derived chaotic state $x(k)$ incorrect. Apparently, the errors are induced by the swapped pixels that correspond to the elements of f_m in S_4. In the following, the probability of such errors, $p_{se} = Prob\left[f_m(l) \in S_4\right]$, is studied. For any swapped pixel $f(l)$ in the k-th pixel-block ($l = 16k+0 \sim 16k+15$), one has

$$p_{se} = Prob\left[f^{(\oplus)}(l) \in S_4^{(\oplus)}\right] \quad (14)$$

where $f^{(\oplus)}(l) = f\left(2\lfloor l/2 \rfloor\right) \oplus f\left(2\lfloor l/2 \rfloor + 1\right)$ and

$$S_4^{(\oplus)} = \Big\{ Seed1(k) \oplus Seed(l), \overline{Seed1(k)} \oplus Seed(l), Seed2(k) \oplus Seed(l), \overline{Seed2(k)} \oplus Seed(l) \Big\}.$$

Considering the Gaussian-like distribution of $f^{(\oplus)}$ (see Fig. 6) and the fact that $0 \in S_4^{(\oplus)}$, p_{se} is generally not negligible for

[2]The 1-value combinations are included since $Seed1(k) = Seed2(k)$ may occur with a small probability.

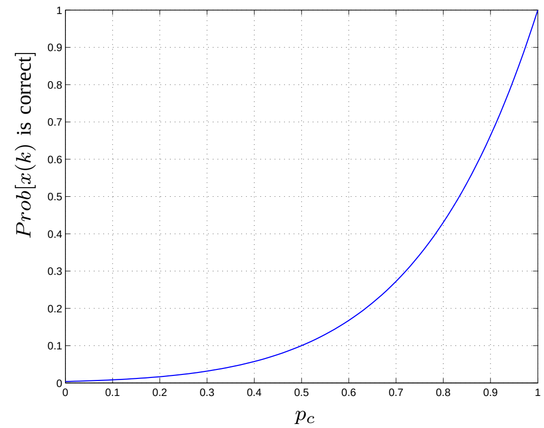


Fig. 10. $P_1 = Prob[x(k)$ is correct] vs. p_c

natural images. Without loss of generality, assume that each bit in $\{b(i)\}$ yields a balanced distribution over $\{0,1\}$ and any two bits are independent of each other. One can deduce

$$P_1 = Prob[x(k) \text{ is correct}] = \sum_{i=0}^{8} p_b(8,i) \cdot p_c^i, \quad (15)$$

where $p_b(8,i) = \binom{8}{i} \cdot 2^{-8}$, which denotes the probability that there are i pairs of swapped pixels, and $p_c = 1 - p_{se}$. The relation between P_1 and p_c is given in Fig. 10.

2) Deriving μ from Two Consecutive Chaotic States: With two consecutive chaotic states, $x(k)$ and $x(k+1)$, the estimated value of the secret control parameter μ will be $\widetilde{\mu}_k = \dfrac{x(k+1)}{x(k)\cdot(1-x(k))}$. Due to the negative influence of quantization errors, generally $\widetilde{\mu}_k \neq \mu$. As known, chaotic maps are sensitive to noise in the initial condition, so an approximate value of μ will generate completely different chaotic states after several iterations, which implies that $\widetilde{\mu}_k$ can not be directly used instead of μ as the secret key. Fortunately, if $|\widetilde{\mu}_k - \mu|$ is small enough, one can exhaustively search in the neighborhood of $\widetilde{\mu}_k$ to find the right value of μ. To verify which guessed value of μ is the right one, one should iterate the Logistic map from $x(k+1)$ until $x(MN/16-1)$, and then check whether or not the corresponding elements in f_m match the calculated chaotic states. Once a mismatch occurs, the current guessed value is discarded, and the next guess will be tried. To minimize the verification complexity, one can check only a number of chaotic states sufficiently far from $x(k+1)$ to eliminate most (or even all) wrong values of $\widetilde{\mu}_k$, and verify the left few ones by checking all chaotic states from $x(k+2)$ to $x(MN/16-1)$.

Now, the concern is when $|\widetilde{\mu}_k-\mu|$ will be small enough to make the exhaustive search practical. In 24-bit fixed-point arithmetic, μ, $x(k)$, and $x(k+1)$ all have 24 binary decimal bits, and the quantization error of $x(k+1)$ can be explained in the following equation:

$$\begin{aligned} x(k+1) &= \left(\mu\cdot x(k)+e'_{x(k+1)}\right)\cdot(1-x(k))+e''_{x(k+1)} \\ &= \mu\cdot x(k)\cdot(1-x(k))+e_{x(k+1)}, \end{aligned}$$

where $e_{x(k+1)}=e'_{x(k+1)}\cdot(1-x(k))+e''_{x(k+1)}\leq e'_{x(k+1)}+e''_{x(k+1)}$. Considering $e'_{x(k+1)},e''_{x(k+1)}<2^{-24}$ for floor/ceil quantization functions and $e'_{x(k+1)},e''_{x(k+1)}\leq 2^{-25}$ for the round function, $e_{x(k+1)}<2^{-23}$ is true in all cases. Then, the quantization error $e_{\widetilde{\mu}_k}=\mu-\widetilde{\mu}_k$ can be estimated as follows:

$$\begin{aligned} e_{\widetilde{\mu}_k} &= \frac{x(k+1)+e_{x(k+1)}}{x(k)\cdot(1-x(k))}-\frac{x(k+1)}{x(k)\cdot(1-x(k))} \\ &= \frac{e_{x(k+1)}}{x(k+1)}\cdot\frac{x(k+1)}{x(k)\cdot(1-x(k))}=\frac{e_{x(k+1)}}{x(k+1)}\cdot\mu \\ &< \frac{2^{-23}}{x(k+1)}\cdot\mu\leq\frac{4}{2^{23}\cdot x(k+1)}=\frac{1}{2^{21}\cdot x(k+1)}. \end{aligned}$$

When $x(k+1)\geq 2^{-n}$ ($n=1\sim 24$),

$$e_{\widetilde{\mu}_k}=\frac{1}{2^{21}\cdot x(k+1)}\leq\frac{2^n}{2^{21}}=2^{n+3}\times 2^{-24}, \qquad (16)$$

which means the size of the neighborhood of $\widetilde{\mu}_k$ for exhaustive search is 2^{n+3}. To make the search complexity practically small in real attacks, $x(k+1)\geq 0.5$ is suggested to derive μ, which occurs with a probability equal to 0.5.

Combining the above analyses, the final complexity of finding two correct consecutive chaotic states, $x(k)$, $x(k+1)$, and the right value of μ, is

$$O\left(\frac{2\times\left(\binom{16}{2}+\binom{16}{1}\right)}{(0.5\times P_1)^2}\times 2^{1+3}\right)=O\left(\frac{17408}{P_1^2}\right), \qquad (17)$$

which is generally much smaller than the complexity of exhaustively searching all possible keys. As a reference value, when $p_c=0.7$, the complexity is about $O\left(2^{17.8}\right)\ll O\left(2^{48}\right)$.

3) A Quick Algorithm to Guess the Two Random Seeds: Following the above-discussed search process, the found correct chaotic states $x(k)$ and $x(k+1)$ will be close to $x(0)$. Considering the occurrence of two consecutive chaotic states larger than 0.5 as a Bernoulli experiment, the mathematical expectation of k will be $\frac{1}{(0.5\times P_1)^2}=\frac{4}{P_1^2}$ [30]. This means that only tens of known plain-pixels[3] are enough for an attacker to break the chaotic map, which is a very desired feature for attackers. However, as an obvious disadvantage, the search complexity to guess the two random seeds is somewhat large. In fact, for each pixel-block, one can only test a few number of possible 2-value (and 1-value) combinations, not all. If this pixel-block looks not good for guessing the two random seeds, simply discard it and go to the next pixel-block. Following such an idea, a quicker algorithm can be designed to find the two random seeds. In this quick-search algorithm, the found correct chaotic states $x(k)$ and $x(k+1)$ may be far from $x(0)$, so the size of the mask image has to be much larger than $\frac{4}{P_1^2}$.

The quick-search algorithm is based on the following observation: the more the unswapped pixels there are in the k-th pixel-block, the more the number of values in S_4 will be in $\{f_m(16k+j)\}_{j=0}^{15}$. Accordingly, define a new sequence $\left\{\widetilde{f}_m(16k+j)\right\}_{j=0}^{15}$ as follows:

$$\widetilde{f}_m(16k+j)=\min\left(f_m(16k+j),\overline{f_m(16k+j)}\right). \qquad (18)$$

Then, the following is also true: the more the unswapped pixels there are in the k-th pixel-block, the more the number of the values in S_2 will be in $\left\{\widetilde{f}_m(16k+j)\right\}_{j=0}^{15}$, where $S_2=\left\{\min\left(Seed1(k),\overline{Seed1(k)}\right),\min\left(Seed2(k),\overline{Seed2(k)}\right)\right\}$.

Therefore, assuming that there are n_k pairs of unswapped pixels in the k-th pixel-block, the following fact is true: if n_k is sufficiently large, the two most-occurring elements in $\left\{\widetilde{f}_m(16k+j)\right\}_{j=0}^{15}$ are the two values in S_2, with a high probability. Then, when can one say that n_k is sufficiently large? In totally 8 pairs of elements, the average number of pairs in S_2 is $N(S_2)=n_k+(8-n_k)\cdot p_{se}$, and the number of other pairs is $N(\overline{S_2})=8-N(S_2)=(8-n_k)\cdot(1-p_{se})$. From a conservative point of view, let $N(\overline{S_2})<\frac{N(S_2)}{2}$, which ensures that the average number of each value in S_2 is larger than the number of all other values in $\{0,\cdots,127\}-S_2$, with a sufficiently high probability. Solving this inequality, one can

[3] For example, even a 10×10 "tiny" image is enough.

get $n_k \geq 6$, yielding $N(\overline{S_2}) \leq 2 < 3 \leq \dfrac{N(S_2)}{2}$.

Based on the above analyses, the quick-search algorithm is described as follows:

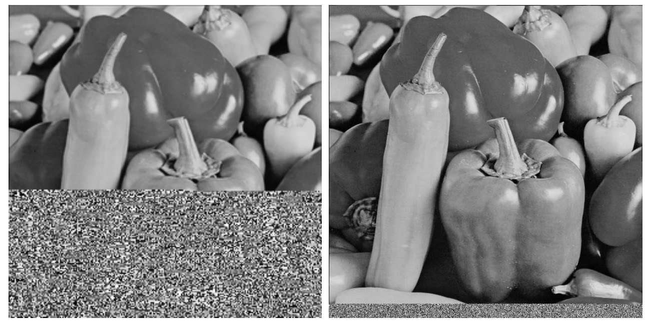
- *Step 1*: find a pixel-block, $\{f_m(16k+j)\}_{j=0}^{15}$, in which the number of values in S_4 is not less than 12;
- *Step 2*: generate a new sequence, $\left\{\widetilde{f}_m(16k+j)\right\}_{j=0}^{15}$;
- *Step 3*: rank all values of $\left\{\widetilde{f}_m(16k+j)\right\}_{j=0}^{15}$ to find the top two mostly-occurring values, $value1$ and $value2$;
- *Step 4*: if the number of $value2$ in $\left\{\widetilde{f}_m(16k+j)\right\}_{j=0}^{15}$ is less than 3, goto *Step 1*;
- *Step 5*: exhaustively search $Seed1(k)$ and $Seed2(k)$ in $\widetilde{S}_4 = \left\{value1, \overline{value1}, value2, \overline{value2}\right\}$.

If more than one value corresponds to the same position in the rank of $\left\{\widetilde{f}_m(16k+j)\right\}_{j=0}^{15}$, all of them should be enumerated as $value1$ and $value2$ in Step 3 to Step 5. In a real attack, some extra constraints can be added to further optimize the above algorithm for different mask images. The attack complexity of this quick-search algorithm is hard to theoretically analyzed, since the distribution of those values that are not in S_4 is generally unknown. Fortunately, experiments show that the complexity is much smaller than the one given above. In Fig. 11, the performance of the quick-search algorithm is shown for the recovered plain-image f^*_{Peppers}, where different pixel-blocks are used to extract the chaotic states. Note that more than forty pixel-blocks are eligible to be used to extract the correct chaotic states, and the three shown here are randomly chosen for demonstration.

In the following, it is theoretically studied as how much MN should be to guarantee the efficiency of the quick-search algorithm, which is determined by the occurrence probability that two consecutive pixel-blocks satisfy the requirements given in Step 1 and Step 3. Assume that each bit in $\{b(i)\}$ yields a balanced distribution over $\{0, 1\}$ and any two bits are independent of each other. The probability that one pixel-block satisfies the requirements, which is denoted by P_o, yields Eq. (19). Then, for the occurrence probability that two consecutive pixel-blocks satisfy the requirements, which is denoted by P_{o2}, one can calculate that $P_{o2} = P_o^2 \geq Prob\left[S_4 = \widetilde{S}_4\right]^2 = \left(\dfrac{4699}{2^{15}}\right)^2 \approx 0.02$. This means that there will be two consecutive pixel-blocks satisfy the requirements in $\dfrac{1}{P_{o2}} \approx 50$ pixel-blocks (about 800 pixels), from the probabilistic point of view. Therefore, the required size of the known plain-image should be larger than 800, which is even smaller than the size

a) The recovered plain-image f^*_{Peppers} from the 7th pixel-block

b) The recovered plain-image f^*_{Peppers} from the 689th pixel-block



c) The recovered plain-image f^*_{Peppers} from the 1673rd pixel-block

d) Recovering a larger plain-image $f^*_{\mathrm{Peppers2},768\times768}$ from the 1673rd pixel-block

Fig. 11. Demonstration of the quick-search algorithm, where ‘Lenna’ is the only known plain-image

$$
\begin{aligned}
P_o \geq Prob\left[S_4 = \widetilde{S}_4\right] &= Prob\left[\text{both } Seed1(k) \text{ and } Seed2(k) \text{ occur at least 3 times in } \left\{\widetilde{f}_m(16k+j)\right\}_{j=0}^{15}\right] \\
&\quad \cdot Prob\left[\min\left(Seed1(k), \overline{Seed1(k)}\right) \neq \min\left(Seed2(k), \overline{Seed2(k)}\right)\right] \\
&= \sum_{n_k=6}^{8}\left(\binom{8}{n_k}\cdot 2^{-8}\cdot\left(1-\sum_{m=0}^{2}\binom{2n_k}{m}\cdot 2^{-2n_k}\right)\cdot\left(1-128^{-1}\right)\right)
\end{aligned} \tag{19}
$$

of a 30×30 image. Hence, the quick-search algorithm is very efficient to use for attacks.

4) Breaking the Chaotic Map with both f_m and Q: All the above-mentioned algorithms are based on only-one known plain-image. When more than one plain/cipher-image is known, the constructed swapping $(0, 1)$-matrix Q will be very useful to increase the efficiency of the attack. As already known, the mask image f_m can be amended using the swapping information stored in Q. Since all amended elements in f_m are also values in S_4, it is obvious that the efficiency of the search algorithm for finding correct random seeds will be increased. In addition, the swapping matrix Q can be used to uniquely determine some bits in $\{b(24k+16+i)\}_{i=0}^{7}$ without checking $f_m(16k + 2i) \in S_4$ and $f_m(16k + 2i + 1) \in S_4$. Thus, the total complexity in finding a correct chaotic state will be less, and the attack will succeed faster.

When two or more plain-images and/or cipher-images are known, most swapped pixels can be successfully distinguished. In this case, it is much easier to find a pixel-block of f_m whose elements are all in S_4, which means that $Seed1(k)$, $Seed2(k)$ can be quickly guessed by enumerating all values in S_4, and all the 8 bits $\{b(24k+16+i)\}_{i=0}^{7}$ can be absolutely determined. This implies that the attack complexity is minimized to be the complexity of breaking RCES's weaker parent – CKBA [24].

D. The Combined Known-Plaintext Attack

The above two known-plaintext attacks have their disadvantages: the first attack cannot decrypt the cipher-images larger than MN (the size of f_m), and the second one cannot decrypt all pixels before the position where the first correct chaotic state $x(k)$ is found. One can combine them, however, to make a better known-plaintext attack without these disadvantages: use the first attack to decrypt the pixels before $x(k)$ and then use the second attack to decrypt the others. Figure 12 shows the performance of this combined attack with only one known plain-image, where the recovered chaotic state in the second attack is selected as $x(1673)$ (see also Fig. 11c), which can make clearer the boundary of the two parts decrypted by the two attacks.

a) The recovered plain-image f_{Peppers} b) The recovered larger plain-image $f_{\text{Peppers2},768\times768}$

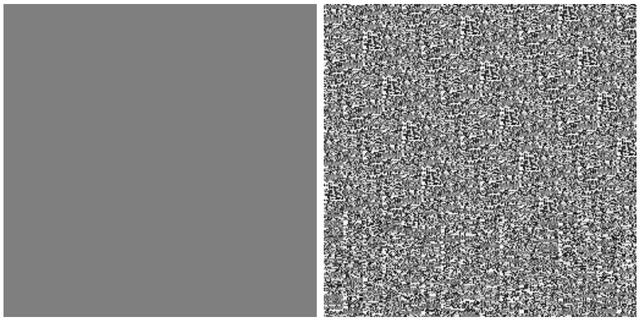
Fig. 12. The recovery performance of the combined known-plaintext attack

E. The Chosen-Plaintext Attack

Apparently, all the above three known-plaintext attacks can be extended to chosen-plaintext attacks.

For the first kind of known-plaintext attack, the chosen-plaintext version can achieve much better recovery performance with a nearly-perfect mask image f_m, by choosing only one plain-image whose pixels are all fixed to be the same gray value. Given such a plain-image, from Corollary 1, any recovered plain-pixel will be the plain-pixel itself or its adjacent pixel. Thus, although the recovery error induced by $a_1 = f_1(16k + 2i) \oplus f_1(16k + 2i + 1)$ may still be large, it is expected that the visual quality of the recovered plain-image will be much better. It is also expected that all salt-pepper impulsive noises will disappear and a dithering effect of edges will occur, which is demonstrated in Fig. 13c with the plain-image f^*_{Peppers} recovered from the chosen plain-image shown in Fig. 13a. As a natural result, the visual quality of the recovered plain-image f^*_{Peppers} becomes much better as compared with the one shown in Fig. 5a.

Similarly to the known-plaintext attack, with some image processing techniques, the recovered plain-image in the chosen-plaintext attack can also be enhanced to further provide a better visual quality. Now, the question is: can one maximize the visual quality with an optimization algorithm? The answer is yes. In fact, with a subtly-designed algorithm, almost all dithering edges can be perfectly polished and a matrix Q containing partial swapping information can be constructed with only one chosen plain-image. In the following, this



a) The chosen plain-image $f_{\rm Gray}$ b) The mask image $f_{{\rm Gray},m}$

c) The recovered plain-image $f^*_{\rm Peppers}$ d) The difference image $\left|f^*_{\rm Peppers} - f_{\rm Peppers}\right|$

Fig. 13. The recovery performance of the chosen-plaintext attack

efficient algorithm and its real performance are studied in some details.

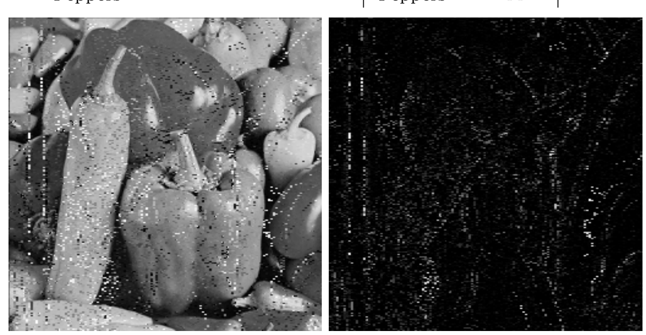
The proposed algorithm divides the image into $2n$-pixel blocks for enhancement, where $2n$ can exactly divide M. The basic idea is to exhaustively search the optimal swapping states of all pixels to achieve the minimal differential errors. For the m-th $2n$-pixel block $f_B(m) = \{f(m \cdot 2n + i)\}_{i=0}^{2n-1}$, the algorithm works as follows:

1) set $\{b_s(i) = 0\}_{i=0}^{n-1}$ and $\Delta_{min} = 256(n-1)$;
2) for $(b_0, \cdots, b_{n-1}) = (\overbrace{0, \cdots, 0}^{n}) \sim (\overbrace{1, \cdots, 1}^{n})$, do
   a) assign $A = \{a_0, \cdots, a_{2n-1}\} = f_B(m)$;
   b) for $i = 0 \sim n-1$, do $Swap_{b_i}(a_{2i}, a_{2i+1})$;
   c) calculate $\Delta A = |a_2 - a_1| + |a_4 - a_3| + \cdots + |a_{2i} - a_{2i-1}| + \cdots + |a_{2n-2} - a_{2n-3}|$;
   d) if $\Delta A < \Delta_{min}$, then set $\Delta_{min} = \Delta A$ and $\{b_s(i) = b_i\}_{i=0}^{n-1}$.
3) for $i = 0 \sim n-1$, do $Swap_{b_s(i)}(f(m \cdot 2n + 2i), f(m \cdot 2n + 2i + 1))$;
4) set the corresponding elements of the swapping matrix Q to be 1 for $b_s(i) = 1$.

The complexity of the above algorithm is $O(2^n \cdot MN)$. When $M = N = 256$ and $n = 8$, it is less than 2^{24}, which is practical even on PCs.

For the recovered plain-image $f^*_{\rm Peppers}$ shown in Fig. 13c, the above algorithm has been tested with parameter $n = 8$, and the result is given in Figs. 14a and 14b. Although the enhanced plain-image have 14378 (about 21.94% of all) pixels different from the original plain-image, its visual quality is so perfect that no any visual degradation can be distinguished. In fact, in a sense, the enhanced plain-image can be considered as a better version of the original one, since each $2n$-pixel block of the former reaches the minimum of the accumulated differential error. From such a point of view, this optimization algorithm can also be used to enhance the visual quality of the plain-image recovered by a known-plaintext attack. For the recovered plain-image shown in Fig. 5a, the enhancing result is given in Figs. 14c and 14d. It can be seen that dithering edges existing in the plain-image shown in Fig. 5a have been polished.

a) Enhancing the plain-image $f^*_{\rm Peppers}$ shown in Fig. 13c b) The difference image $\left|f^*_{\rm Peppers} - f_{\rm Peppers}\right|$ of a)



c) Enhancing the plain-image $f^*_{\rm Peppers}$ shown in Fig. 5a d) The difference image $\left|f^*_{\rm Peppers} - f_{\rm Peppers}\right|$ of c)

Fig. 14. The performance of the optimization algorithm when $n = 8$

In the above algorithm, most swapped operations can be distinguished by using the minimum-detecting rule on the accumulated differential error of $f_B(m)$, which means that most elements in Q are correct for showing the real values of the swapping directive bits $\{b(24k+16+i\}_{i=0}^{7}$. Once 32 consecutive correct elements (two 16-pixel blocks) in Q have been found, it is possible to derive μ and a chaotic state $x(k)$, like in the situation of the second known-plaintext attack.

V. Lessons Learned from RCES/CKBA

From RCES and its parent version CKBA, some principles can be suggested for the design of good image encryption schemes. Although the security of RCES and CKBA against the known/chosen-plaintext attack is very weak, they are still useful as typical carelessly-designed examples to show what one should do and what one should not do[4].

A. Principle 1: Security against the known/chosen-plaintext attacks should be provided

As surveyed in [2], besides CKBA/RCES, many other image encryption schemes are also insecure against the known/chosen-plaintext attack. However, without the capability against the known/chosen-plaintext attacks, it will be insecure to repeatedly use the same secret key to encrypt multiple image files. When the cryptosystems are used to encrypt image streams transmitted over networks, this problem can be relaxed due to the use of time-variant session keys [27]. Considering that most image encryption systems are proposed to encrypt local image files, the security against the known/chosen-plaintext attacks is generally required.

B. Principle 2: Do not use invertible encryption function

Rewrite the encryption function of a symmetric cipher as $C = E(P, K)$. The function $E(\cdot, \cdot)$ is said to be invertible, if K can be derived from C and P with its inverse function $E^{-1}(\cdot, \cdot)$, i.e., $K = E^{-1}(P, C)$. Most modern ciphers employs a mixture of operations defined in different groups to make the encryption function non-invertible.

[4]For more discussions on how to design a good image encryption schemes, see Sec. 4.5 of [2].

In RCES/CKBA, the encryption function is XOR, which is an invertible operation since $P \oplus K = C \Rightarrow K = P \oplus C$. It is the essential reason why the mask image f_m can be used as an equivalent of the real key $(x(0), \mu)$. Similarly, the invertibility of the swapping operations is the reason for the success of the dithering-removal algorithm discussed in the chosen-plaintext attack.

To enhance the security of RCES, the XOR operation can be replaced with some key-dependent invertible functions. Another way is to replace the swapping operation with more complex long-distance permutation operations, such as the ones used in [9]–[12]. If both operations are changed as above, the security will be further enhanced. References [9]–[12] suggest some typical image ciphers that use such an idea to ensure the security against the known/chosen-plaintext attacks.

C. Principle 3: The correlation information within the plain-image should be sufficiently reduced

As shown in the previous section, the high correlation information between adjacent pixels is an important reason of the good performances for the known/chosen-plaintext attacks. In fact, there exists a large amount of correlation information within digital images, even between pixels whose distances are large, such as pixels in a smooth area. To provide sufficient security against attacks, the correlation information within the plain-image should be sufficiently concealed. A typical method to conceal the correlation information is to carry out complex long-distance permutation operations [9]–[12]. Note that the long-distance permutations are not necessary conditions, but sufficient ones, since any secure text cipher can also provide enough security for digital images.

D. Principle 4: Any non-uniformity existing in the cipher-images should be avoided

From a cryptographer's point of view, any non-uniformity is not welcome due to the risk of causing statistics-based attacks, such as the well-known differential attacks [27]. So, it should be carefully checked whether or not there exists any non-uniformity in the ciphertexts.

The essential reason for the insecurity of RCES/CKBA against the known/chosen-plaintext attacks can also be ascribed to the non-uniformity of the distribution of $f(l) \oplus f'(l)$ over $\{0, \cdots, 255\}$:

- for any unswapped pixel, $Prob[f(l) \oplus f'(l) = Seed(l)] = 1$, i.e., the distribution is one with the most non-uniformity;
- for any swapped pixel, the distribution of $f(l) \oplus f'(l)$ has the same non-uniformity level as the one of $f(l) \oplus f(l+1)$ (see the distribution of $f_{\rm Peppers}^{(\oplus)}$ shown in Fig. 6).

This also suggests that all pixels should be permuted. Actually, in the second known-plaintext attack, the feasibility of the quick-search algorithm in finding the two random seeds is benefited from the non-uniformity of the distribution of $\left\{\widetilde{f}_m(16k+j)\right\}_{j=0}^{15}$ over the discrete set $\{0, \cdots, 127\}$. If each $\widetilde{f}_m(16k+j)$ distributes uniformly over $\{0, \cdots, 127\}$, the exhaustive search algorithm will be practically impossible when the block size is changed to a sufficiently large value.

VI. Conclusion

In this paper, it has been pointed out that the RCES/RSES image encryption method recently proposed in [17], [18] is not secure enough against the known/chosen-plaintext attacks, and that the security against brute-force attack was overestimated. Both theoretical and experimental analyses have been given to support the feasibility of the known/chosen-plaintext attacks. The insecurity of RCES are caused by a careless design, and some principles on good design of secure image encryption schemes can be learned from the weakness of RCES. In summary, although RCES cannot be used in practice as a secure cipher to protect digital images, it provides a typical example for caution.

Acknowledgement

This research was partially supported by the Applied R&D Centers of the City University of Hong Kong under grants no. 9410011 and no. 9620004, and by the National Natural Science Foundation, China, under grant no. 60202002. The second author would like to thank Miss Min Huang at Zhejiang University for her comments on part of the theoretical analyses of the designed attacks presented in this paper.

References

[1] P. P. Dang and P. M. Chau, “Image encryption for secure internet multimedia applications,” *IEEE Trans. Consumer Electron.*, vol. 46, no. 3, pp. 395–403, 2000.

[2] S. Li, G. Chen, and X. Zheng, “Chaos-based encryption for digital images and videos,” in *Multimedia Security Handbook*, B. Furht and D. Kirovski, Eds. CRC Press, LLC, April 2004, the preprint is available at http://www.hooklee.com/pub.html (and also at http://www.ee.cityu.edu.hk/~gchen/Internet.htm).

[3] N. G. Bourbakis and C. Alexopoulos, “Picture data encryption using SCAN patterns,” *Pattern Recognition*, vol. 25, no. 6, pp. 567–581, 1992.

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

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

[6] H. Cheng and X. Li, “Partial encryption of compressed images and videos,” *IEEE Trans. Signal Processing*, vol. 48, no. 8, pp. 2439–2451, 2000.

[7] C.-C. Chang, M.-S. Hwang, and T.-S. Chen, “A new encryption algorithm for image cryptosystems,” *J. Systems and Software*, vol. 58, no. 2, pp. 83–91, 2001.

[8] A. Pommer, “Selective encryption of wavelet-compressed visual data,” Ph.D. dissertation, Department of Scientific Computing, University of Salzburg, Austria, June 2003.

[9] S. S. Maniccam and N. G. Bourbakis, “Image and video encryption using SCAN pattern,” accepted by *Pattern Recognition* in 2003, available online at http://www.sciencedirect.com on 5 Dec. 2003.

[10] J. Scharinger, “Fast encryption of image data using chaotic Kolmogorov flows,” *J. Electronic Imaging*, vol. 7, no. 2, pp. 318–325, 1998.

[11] J. Fridrich, “Symmetric ciphers based on two-dimensional chaotic maps,” *Int. J. Bifurcation and Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998.

[12] Y. Mao, G. Chen, and C. K. Chui, “A novel fast image encryption scheme based on 3D chaotic Baker maps,” accepted by Int. J. Bifurcation and Chaos in June 2003.

[13] K. Yano and K. Tanaka, “Image encryption scheme based on a truncated Baker transformation,” *IEICE Trans. Fundamentals*, vol. E85-A, no. 9, pp. 2025–2035, 2002.

[14] J.-C. Yen and J.-I. Guo, “A new image encryption algorithm and its VLSI architecture,” in *Proc. IEEE Workshop Signal Processing Systems*, 1999, pp. 430–437.

[15] ——, “Efficient hierarchical chaotic image encryption algorithm and its VLSI realisation,” *IEE Proc. – Vis. Image Signal Process.*, vol. 147, no. 2, pp. 167–175, 2000.

[16] ——, "A new chaotic key-based design for image encryption and decryption," in *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS'2000)*, vol. 4, 2000, pp. 49–52.

[17] H.-C. Chen, J.-C. Yen, and J.-I. Guo, "Design of a new cryptography system," in *Proc. PCM'2002*, ser. Lecture Notes in Computer Science, vol. 2532. Springer-Verlag, Berlin, 2002, pp. 316–329.

[18] H.-C. Chen and J.-C. Yen, "A new cryptography system and its VLSI realization," *J. Systems Architecture*, vol. 49, pp. 355–367, 2003.

[19] J.-K. Jan and Y.-M. Tseng, "On the security of image encryption method," *Information Processing Letters*, vol. 60, no. 5, pp. 261–265, 1996.

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

[21] H. C. H. Cheng, "Partial encryption for image and video communication," Master's thesis, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, Fall 1998.

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

[23] S. Li and X. Zheng, "On the security of an image encryption method," in *Proc. IEEE Int. Conference on Image Processing (ICIP'2002)*, vol. 2, 2002, pp. 925–928.

[24] ——, "Cryptanalysis of a chaotic image encryption method," in *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS'2002)*, vol. II, 2002, pp. 708–711.

[25] S. Li, "Analyses and new designs of digital chaotic ciphers," Ph.D. dissertation, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China, June 2003, available online at http://www.hooklee.com/pub.html (in both Chinese and English).

[26] S. Li, C. Li, G. Chen, and X. Mou, "Cryptanalysis of a hierarchical chaotic image encryption," in preparation, to be submitted in 2004.

[27] B. Schneier, *Applied Cryptography – Protocols, algorithms, and souce code in C*, 2nd ed. New York: John Wiley & Sons, Inc., 1996.

[28] B.-L. Hao, *Starting with Parabolas: An Introduction to Chaotic Dynamics*. Shanghai, China: Shanghai Scientific and Technological Education Publishing House, 1993, (In Chinese).

[29] T. Kohda and K. Aihara, "Chaos in discrete systems and diagnosis of experimental chaos," *Trans. IEICE*, vol. E 73, no. 6, pp. 772–783, 1990.

[30] The editorial committee of Modern Applied Mathematics Handbooks, *The Handbook of Probability Theory and Stochastic Process*, ser. Modern Applied Mathematics Handbooks. Beijing, China: Tsinghua University Press, July 2000, vol. 3, (In Chinese).