Security Analysis and Improvement on an Image Encryption Algorithm Using Chebyshev Generator

Tsu-Yang Wu[1,2,3], Xiaoning Fan[4], King-Hang Wang[5], Jeng-Shyang Pan[2,3], Chien-Ming Chen[4]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, China
[3]National Demonstration Center for Experimental Electronic Information and Electrical Technology Education, Fujian University of Technology, China
[4] Harbin Institute of Technology (Shenzhen), China
[5] Department of Computer Science and Engineering, Hong Kong University of Science and Technology
wutsuyang@gmail.com, 1203119830@qq.com, kevinw@cse.ust.hk,
jengshyangpan@fjut.edu.cn, chienming.taiwan@gmail.com

Abstract

Chaotic map including Chebyshev's polynomial have been studied and used in many cryptographic areas recently due to its low cost of computation and high level of security. Some research works have been proposed to use Chebyshev's polynomial in image encryption by setting up two-stage encryption algorithms. Pixels of a plain image are first permuted by a Permutation process. Then each pixel values are changed by a Diffusion process. A two-stage image encryption algorithm is generally believed to be more secure than a single stage image encryption algorithm.

In this paper, however, we demonstrate a recent two-stage image encryption algorithm proposed by Wang et al., is insecure against chosen plaintext attack. An attacker may be able to decrypt a cipher image after knowing some ciphers of images which are chosen by the attacker. We present an subtle but efficient improvement over Wang et al.'s algorithm so that it is not only immune to the attack we presented but also statistically improved when experiment is conducted to measure pixels' correlation, NPCR and UACI.

Keywords: Image encryption, Chaotic map, Security, Chosen plaintext attack

1 Introduction

Traditional symmetric key encryption algorithm DES and AES are designed to encrypt general purpose content and have been widely used in the world. With the fast growth of image technologies, researchers are still putting their efforts in designing new algorithm to protect digital image from piracy, counterfeiting, or simply data secrecy. This is due to the fact that image encryption usually involves a large storage capacity and a high correlation among pixels, therefore the general purpose encryption scheme are too slow or consume too many storage while not achieving a significant better security.

Some recent works [1-18] studied the possibility of applying chaotic maps on image encryption. There are some nice properties, such as ergodicity, pseudo-randomness, and sensitivity, from the chaotic map hinting that it is a good candidate of primitives for constructing an image encryption algorithm.

Recently researchers have been studying the structure of two-stage encryption to further decorrelate the adjacent pixels and to enhance image security. A two-stage encryption algorithm can be roughly understood as that pixels of a plain image is first permuted and then modified according to key and the previous processed pixel. In 2012, Huang [6] proposed an efficient chaotic image encryption algorithm that first generates two pseudorandom chaotic sequences by a discrete Chebyshev function. The two sequences are used to decorrelate adjacent pixels in a permutation process. Then, a pseudorandom chaotic sequence is generated by a two-dimensional Chebyshev function in diffusion process. Recently, Wang et al. [8] pointed out Huang's image encryption algorithm is insecure against a chosen plaintext attack. Meanwhile, they also proposed a modification based on his encryption algorithm so that they claim their algorithm would be secure under a chosen plaintext attack model. In this paper, however, we find Wang et al.'s encryption algorithm is vulnerable against chosen plaintext and chosen ciphertext attacks. In order to overcome the two attacks, we propose an improvement based on Wang et al.'s algorithm. The simulation results are shown that our improved image encryption algorithm has a better efficiency than Wang et al.'s algorithm.

The remainder of this paper is organized as follows.

*Corresponding Author: Chien-Ming Chen; E-mail: chienming.taiwan@gmail.com
DOI: 10.3966/160792642019012001002

In Section 2 we present the background of the algorithm that includes Chebyshev's polynomial, some related works, and our adversary model. It is followed by a review of Wang et al.'s image encryption algorithm in Section 3. Then we present security flaws of Wang et al.'s algorithm under a chosen plaintext attack and our improvement in Section 4 and 5 respectively. Some experiment results and analyses are given in Section 6 and a conclusion is given in Section 7.

2 Background

2.1 Chebyshev Polynomial

Many crypto research including Wang et al.'s adopt Chebyshev chaotic function in their design. It is because that a Chebyshev chaotic function provides three good properties:

1. Ergodicity. Given a fixed domain, the chaotic function will traverse the entire corresponding range within a finite time.
2. Sensitivity. For an arbitrarily small change or perturbation, the result of chaotic function may output significantly different values.
3. Pseudo randomness. This property is coming from the ergodicity and the sensitivity.

The Chebyshev chaotic function, sometime also referred as the Chebyshev polynomial, of the order n [19-27] is recursively defined over real number by

$$T_n(x) = \begin{cases} 1, & if\ n = 0 \\ x, & if\ n = 1 \\ 2x \cdot T_{n-1}(x) - T_{n-2}(x), & if\ n > 2 \end{cases} \quad \textbf{(1)}$$

where $T_n(x), x \in [-1,1]$. It can also be expressed as $T_n(x) : [-1,1] \to [-1,1]$ by

$$T_n(x) = \cos(n \arccos x)\ for\ n = 0,1,... \quad \textbf{(2)}$$

A two dimensional Chebyshev function [28] is recursively defined by

$$\begin{cases} x_i = 1 - \alpha \cdot y_{i-1}^2 \\ y_i = \cos(\beta \cdot \arccos x_{i-1}) \end{cases} \quad \textbf{(3)}$$

for given two control parameters α, β and initial values x_0 and $y_0 \in (-1, 1)$. This two dimensional function allows us to process on a 2D image easier later.

2.2 Related Works

Our research is aiming at presenting a vulnerability of Wang et al. [8] which is motivated by the philosophy of Huang's research [6] which has been cracked by Wang et al in their paper. We will give a more detail description of Wang et al.'s scheme in a later section. Here we present some relevant works in the literature.

Chaotic maps have been received more and more attentions from research community when designing image encryption algorithms. Some literature [2, 5, 7, 10] adopt diffusion process only in their encryption algorithms. Behnia et al. [2] proposed a novel image encryption algorithm based on coupled maps. It combines one-dimensional and two-dimensional chaotic maps to generate random sequences for hiding the pixel values of the image. Hussian et al. [5] proposed an image encryption algorithm with combining a NCA map [29] and a S8 S-box [30]. However, Zhang and Xiao [7] showed that Hussian et al.'s algorithm is insecure against chosen plaintext attack. They concluded that S-box-only image ciphers cannot be secure against chosen plaintext attacks.

To break through the security limits of diffusion-only schemes, researchers [1, 3-4, 6, 8-9, 11] adopt both permutation (confusion) and diffusion processes with two separated encryption keys to strengthen their encryption systems. Gao and Chen [1] used two shuffling vectors generated by a Logistic map to scramble the rows and columns of image pixels and then encrypted the shuffling image by a hyper-chaos system. However, Rhouma and Belghith [3] found their algorithm is insecure against chosen plaintext and chosen ciphertext attacks and proposed their improvement scheme. Jeng et al. [11] later pointed out that the Rhouma-Belghith improvement and the Gao-Chen algorithm have low sensitivity towards the changes in the plain image which suggested their schemes are vulnerable to chosen-plaintext attacks. On a separated branch Liu and Wang [4] proposed a chaotic color image encryption algorithm. In their encryption process only each RGB channel is encrypted separately using a key generated by a piecewise linear Chebyshev map.

In the work by Wang and Guo [9], a chaotic sequence is generated by a Logistic map to shuffle the position of plain image. This encryption process is repeated for many rounds in order to decorrelate pixels. However, Yap et al. [12] found that the Wang and Guo's encryption algorithm [9] is insecure against the differential attack where then two chosen images are encrypted the differential information of these two cipher will allow the attacker to break the system. Such a technique is also employed in our paper, but in a different way. Xu et al. [15] presented an algorithm that divides the plain image is into two sequences for processing. This idea is also adopted in Huang [6], the trunk of our research branch. Later Wang et al. [13], Wang et al. [14], and Ye and Huang [16] proposed their own schemes that use different technologies like Logistic map, Tent map, coupled map lattice, DNA sequence operation, 3D cat map, and SHA-3 hash function.

As we can see many of the recent research mentioned above share some similar properties:

1. Using both confusion and diffusion processes in

their algorithm design.

2. Using two separated key generated by a chaotic map.

3. Recognizing the importance of chosen plaintext attacks and trying to avoid it in their design, regardless if they can avoid it or not.

We presented the differential attacks on a particular encryption algorithm in this paper while we did not deny the possibility that our attack can also be applied on some other algorithms that share the same properties. In fact we are working toward to explore such possibility.

2.3 Attacker Models

The terms Chosen Plaintext Attacks (CPA), one-wayness (OW), indistinguishable (IND) are commonly used in the crypto-community to describe different models. CPA describes the capability of an adversary while OW and IND describes different goals of the adversary. We briefly describe them in this section. Readers may take the following references [3, 7-8, 11, 31-33] if they are keen to explore further.

Chosen Plaintext Attacks. In this attack model, we assume that adversary A has obtained a temporary access of encryption machinery E. Then, A can choose a set of plain images P to construct the corresponding cipher images C. The adversary shall use this machinery as a black-box without knowing the internal states of it nor the key. Then, the adversary will be challenged with an encrypted image. With the challenge the attacker needs to complete a specific goal. This goal could be OW or IND.

One-wayness (OW). With the given challenge the adversary would need to produce the corresponding plaintext.

Indistinguishable (IND). He will be required to differentiate the plain image of the encrypted image and a random image with the same size.

We name OW-CPA or IND-CPA for a chosen plaintext attacker who need to perform OW or IND respectively. When we say it is secure against OW-CPA or IND-CPA that means no polynomial time adversary could achieve their goal with CPA capability.

3 Review of Wang et al.'s Image Encryption Algorithm

The notations used in Wang et al.'s algorithm [8] are summarized in Table 1 and the flowchart of their algorithm is depicted in Figure 1. Wang et al.'s algorithm consists of three main processes: Sequences Generation, Permutation, and Diffusion. Briefly speaking, in the process Sequences Generation two set of encryption sequences are generated. One set as denoted by (H, L), is used in the permutation process and is referred as the permutation key. The other set, as denoted by $\{\mu_i\}_{i=1}^{mn}$, is used in the diffusion process and is referred as the diffusion key. In the Permutation process, a 2D monocolor image I will be scrambled by the permutation key resulting a 2D monocolor image I' such that the position of each pixel will be different. The image will be transformed into a 1D vector v and processed by the diffusion process. The diffusion process will encrypt each pixel using the diffusion key and resulting a cipher vector c'. By rearranging the cipher vector it becomes a cipher image c.

Table 1. Table of Notations

Notation	Meanings
I	An 8-bit gray image with size $m \times n$
C	A cipher image with the same size as the plain image.
$(\bar{x}_0, \hat{x}_0)$	The encryption key for the Permutation process.
(x_0, y_0)	The encryption key for the Diffusion process.
p, q	System parameters for the Permutation key.
r	A system parameter for the Diffusion key.
H, L	The Permutation key – encryption sequences for the Permutation process.
$\{u_i\}_{i=1}^{mn}$	The Diffusion key – an encryption sequence for the Diffusion process.
$x \bmod y$	The remainder of x divided by y.
c_0, t	Two arbitrary constants used in the Diffusion process.
$g(\cdot)$	Any simple function, for example, $g(x) = x$.
$x << y$	A bitwise cyclic left shift operation on x by y-bits.

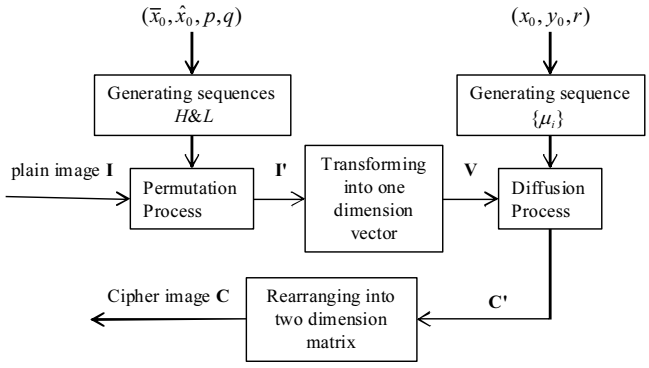


Figure 1. The flowchart of Wang et al.'s algorithm

3.1 Details of the Wang et al.'s Scheme

1. Generating Pseudo-random sequences.

- Inputs: The encryption key $(\bar{x}_0, \hat{x}_0, x_0, y_0)$ and the system parameters (p, q, r)
- Outputs: Two sets of encryption sequences: the permutation key (H, L) where H is size m, L with size n, and the diffusion key $\{u_i\}$ with size $m \times n$.
- Procedures - Generating the Permutation Key:

(a) The process computes two chaotic sequences $\{\bar{x}_i\}_{i=0}^{\infty}$ *and* $\{\hat{x}_i\}_{i=0}^{\infty}$ using the following iterations with the inputted initial values $\bar{x}_0$ *and* $\hat{x}_0$.

$$x_i = 8x_{i-1}^4 - 8x_{i-1}^2 + 1, \quad \textbf{(4)}$$

(b) It defines two new sequences $\{\overline{x}_i'\}_{i=1}^{m+n} = \{\overline{x}_i'\}_{i=p+1}^{m+n+p}$ and $\{\hat{x}_i\}_{i=1}^{m+n} = \{\hat{x}_i\}_{i=q+1}^{m+n+q}$. Then, dividing the sequence $\{\overline{x}_i'\}_{i=1}^{m+n}$ into two parts $P_1 = \{\overline{x}_i'\}_{i=1}^{m}$ *and* $P_2 = \{\overline{x}_i'\}_{i=m+1}^{m+n}$. Similarly, the sequence $\{\hat{x}_i'\}_{i=1}^{m+n}$ is divided into $Q_1 = \{\hat{x}_i'\}_{i=1}^{n}$ and $Q_2 = \{\hat{x}_i'\}_{i=n+1}^{m+n}$.

(c) P_2 and Q_2 are then sorted ascendingly and the original position of each terms, becomes the sequences S_1 and S_2. So for example if P_2 = {0.4, –0.2, 0.3, 0.5}, S_1 would be {2, 3, 1, 4}. Note the length of S_1 and S_2 would be n and m respectively. Each element in S_1 and S_2 would be a unique integer in the range [1, *n*] and [1, *m*] respectively. Then, it reorders the sequences Q_1 and P_1 according S_1 and S_2 respectively, the resulted sequences are called Q_1' *and* P_1'.

(d) Two sequences Q_1' *and* P_1' are then sorted again ascendingly and the original position of each terms becomes the sequences *H* and *L* respectively. Note that *H* has *m* unique elements from [1, *m*] and *L* has *n* unique elements from [1, *n*].

• Procedures - Generating the Diffusion Key:

(a) Assume *r* is a defined system parameter, a chaotic sequence

$$\{w_i\}_{i=1}^{mn} = \{x_{r+1}, y_{r+1}, x_{r+2}, y_{r+2}, \cdots, x_{mn/2}, y_{mn/2}\}$$

is defined using the two-dimensional Chebyshev iteration as follows:

$$\begin{cases} x_i = 1 - 2 \cdot y_{i-1}^2 \\ y_i = \cos(6 \cdot \arccos x_{i-1}) \end{cases} \quad \textbf{(5)}$$

where x_0 and y_0 are two secret keys. Note that the sequence $\{w_i\}$ has a length *mn*.

(b) We output the diffusion key as the sequence $\{\mu_i\}_{i=1}^{mn}$ by the following:

$$\mu_i = \text{mod}(\lfloor w_i \times 10^{14} \rfloor, 256). \quad \textbf{(6)}$$

2. Permutation process.

(a) Inputs: An 8-bit gray color *m*-by-*n* pixels plain image *I* and the Permutation keys *H* and *L*.

(b) Outputs: An 8-bit gray color *m*-by-*n* pixels scrambled image *I*′.

(c) Procedures: The image I is treated as 2D array. For each row *i* in I will be moved to a new row according to the *i*-th element in *H*. On each column *j* of the resulting array will be moved to a new column according to the *j*-th element in *L*. The shifted array is assigned as the scrambled image *I*′.

3. Diffusion process.

(a) Inputs: The Diffusion key $\{\mu_i\}_{i=1}^{mn}$ and a length-*mn* 1D vector *v* transformed from the scrambled image *I*′, by scanning it from top to bottom, left to right.

(b) Outputs: A length-*mn* 1D vector *c*′ which can be transformed back to an encrypted image *c* by filling the pixel with the value in *c*′, from top to bottom, left to right.

(c) Procedures: For each $v_i \in v$ for i = 1, 2,…, *mn*, computes

$$c_i = (v_i + t \cdot u_i + c_{i-1}' \mod 256) \oplus \mu_i \quad \textbf{(7)}$$

and

$$c_i' = (g(c_i) \oplus \mu_i) << \text{mod}(\mu_i \ mod \ 8) \quad \textbf{(8)}$$

where *t* and c_0 are two arbitrary constants and g is an arbitrary simple function. It returns $c' = \{c_1', c_2', \ldots, c_{mn}'\}$.

4 Cryptanalysis on Wang et al.'s Algorithm

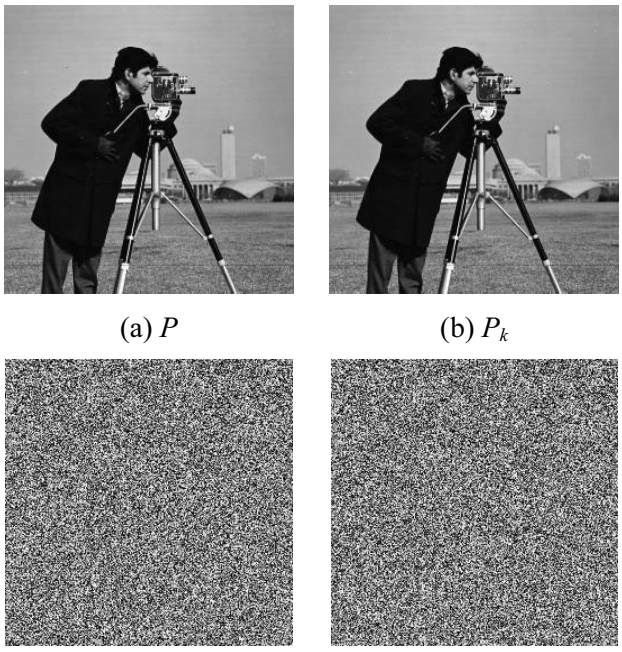
In this section, we show that Wang et al.'s image encryption algorithm is insecure against a chosen plaintext attack. As we are going to illustrate in this attack, the attacker may recover the sequences *H* and *L* after requesting at most max(*m*, *n*) number of chosen plaintext. Then, using *H* and *L* the attacker can recover all unencrypted pixels from a challenge image c.

4.1 Recover Sequences *H* and *L*

We depict the algorithm for recovering the Permutation key in Algorithm 1. The idea is to construct a pair of twins images with one pixel difference. In the Permutation process this pixel will be shifted to another coordinate, says (*i*, *j*). Since an image is encrypted sequentially in the Diffusion process, the pixels before (*i*, *j*) of the twins images would be identical and diffused at (*i*, *j*) onwards. That implies *(i*, *j*) is the starting point of the differentiation. That would reveal the value of one element of the *H* and *L* permutation keys. Iterating the process through the size of the permutation keys (depends which one is larger) would allow us to recover the entire key pair.

For better illustration, we demonstrate one iteration (k = 55) of the attack on a square size image as an example. Two plain images *P* and P_k with only one pixel value different at the position (55, 55) are constructed, as shown in Figure 2(a) and Figure 2(b), where *P* is set 100 and P_k is set 200. The attacker requests the encryption of *P* and P_k from the encryption oracle and produced *C* and C_k respectively, as shown in Figure 2(c) and Figure 2(d).

Computing $C \oplus C_k$ and then scanning the image from top to bottom, left to right, to search the first non-zero element. The element is at (107, 3) as shown in Figure 3. Thus, we can find the 55-th row is permuted to the 107th row and the 55-th column is permuted to the 3th column, say $H(107) = 55$ and $L(3) = 55$. In Table 2, we demonstrate some recovered values of *H* and *L*.



(a) P (b) P_k

(c) C (d) C_k

Figure 2. The twins images P and P_k created with only the position (55, 55) and their cipher images C and C_k

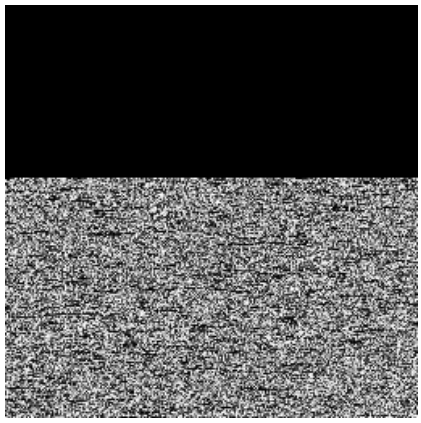


Figure 3. $C \oplus C_k$. It is observed that pixels have a value of zero (black) before the position (107; 3)

Table 2. Array $H[]$ and $L[]$ after executing the 55-th iteration

index	1	…	54	55	56	…	256
$H[]$	43	…	242	107	-	…	-
$L[]$	2	…	68	3	-	…	-

Algorithm 1. Recovering H and L

procedure Recovering(Encryption oracle: O, Image dimension: m, n)
  Define size m integer array H, size n integer array L.
  $P \leftarrow$ random image with size m-by-n.
  $C \leftarrow O(P)$ ◁ Request the cipher image C of P.
  for $k \leftarrow 1, \max(m, n) - 1$ do
    $k_m \leftarrow k \bmod m$, $k_n \leftarrow k \bmod n$.
    $P_k \leftarrow P$
    Change the pixel (k_m, k_n) of P_k to a random value.
    $C_k \leftarrow O(P_k)$ ◁ Request the cipher image Ck of P_k.
    $(i, j) \leftarrow$ the coordinate of the first non-zero element of $C_k \oplus C$.
    $H[k_m] \leftarrow i$, $L[k_n] \leftarrow j$.
  end for
  Fill the last blank cell in $H[]$ and $L[]$ by the missing value.
  return $H \leftarrow H[]$, $L \leftarrow L[]$.
end procedure

4.2 Breaking the Indistinguishability

In the IND-CPA model the attacker should be unable to distinguish which a cipher image is encrypted from either of the two same size images. We present a way for the attacker to break this indistinguishability.

Let I be a 2D array representing a plain image. We define $I' = \pi(I)$ where π a permutation of the elements according to the Permutation process and I' is an 2D array representing the scramble image. We denote $I[y][x]$ (respectively $I'[y][x]$) as the y-th row, x-th column pixel of the image I (respectively I'). We use the symbol $'$ to denote the new position of a pixel after the Permutation process, that is, a pixel of an image I at the coordinate (i, j) will be shifted to the coordinate (i', j') after the Permutation process. In other words, $I[y][x] = I[y'][x']$ for every possible y and x.

Given the H and L obtained in Subsection 4.1, the attacker would be feasible to compute the function π and its inverse π^{-1} and also (y', x') for every pair (y, x). The attacker creates 256 scrambled images I'_i for $i \in [0, 255]$, so that points on these images are random except the pixel $I'_i[1][1] = i$. In other words, the first pixel of this set of scrambled images should be takes all possible values from 0 to 255. The attacker requests the encryption of all plain images Ii, which can be computed by $\pi^{-1}(I'_i)$, from the encryption oracle. The first bit outputted from the Diffusion process, c'_1, depends on the Diffusion key, constants, and the first pixel of the scrambled images only. Therefore the first

bit of the cipher of these images should be all different. A table storing i and the corresponding c_1' are built. This table should contains exactly 256 entries.

Then the attacker is given a challenge cipher image and two plain images to decide which one produces the cipher image. What the attacker needs to do is to scramble the two plain images and look at the first pixels of them. From the table just created the attacker find the corresponding value c_1' and compare that with the first pixel of the cipher image.

By this the attacker would be able to distinguish which plain image produces the cipher text.

This algorithm fails only when the first pixel of the scrambled version of the two plain images are the same, with probability 1/256. And the algorithm can be extended to enumerate all first k-bits of the scrambled images with the cost of 256^k oracle queries where the failure probability would be suppressed to $1/256^k$.

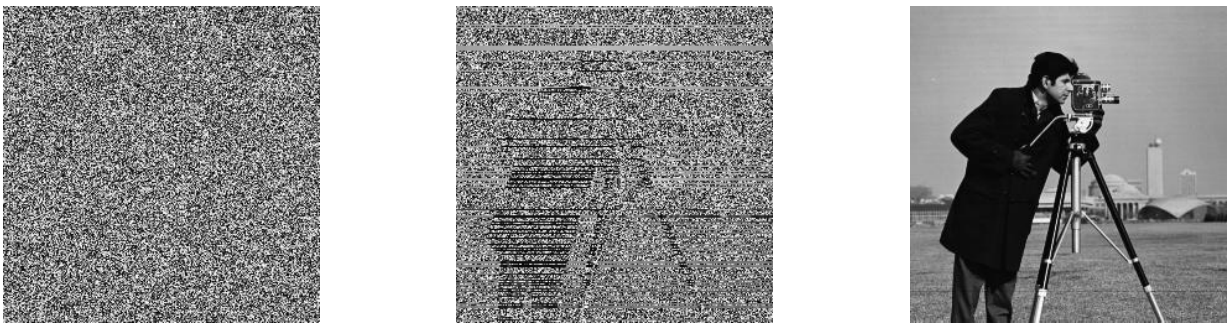
We declare the insecurity of Wang et al. algorithm while we only need not more than $\max(m, n) + d^k$ of non-adaptive encryption oracle queries in total to break their algorithm where m, n are the dimension of the image, d is the color depth of the image ($d = 256$ in this case)with success rate $1 - 1/d^k$.

4.3 Completely Decrypting a Cipher Image

In the above section we have already demonstrated their algorithm is insecure with a small number of encryption oracle queries. In this section we demonstrate how to decrypt a cipher image with a chosen plaintext attack, i.e., proving the system in not OW-CPA secure.

Again we assume the permutation key is obtained and the permutation function π is computable by the attacker. The basic idea of the attack is to match each pixel of the encrypted images by querying some prepared images. Since the permutation is known and the encryption of each pixel depends on the key and a previous pixel, the adversary can recover one pixel with at most 256 queries. We describe the attack in Algorithm 2.

Figure 4 demonstrates a snapshot of reconstructed images using the above algorithm.



(a) Cipher image c' (b) Intermediate result (c) Final result

Figure 4. Snapshot of reconstructed images. The first on shows the cipher image. The second one shows the intermediate result when $y = x = 50$. The final one shows the complete decrypted image which is the same as the plain image

Algorithm 2. Decrypting a cipher image

```
procedure Decrypting(Cipher: c′, Encryption oracle: O, Permutation Function π)
  Define size m-by-n integer array p.
  for y ← 1, m do
    for x ← 1, n do
      for i ← 0, 255 do
        I′_i ← p
        I′_i [y][x] ← i
        C ← O(π^{-1}( I′_i ))          ◁ π^{-1}( I′_i ) unscrambles I′_i .
        if C[y][x] = c′[y][x] then
          p[y][x] ← i
          Continue in x-loop.
        end if
      end for
    end for
  end for
  return I ← p.
end procedure
```

5 Our Improvement and Simulation Results

5.1 Our Improvement

Based on Wang et al.'s image encryption algorithm we propose an improvement to overcome the mentioned weaknesses. Learnt from the above analysis their algorithm is vulnerable against differential attacks. We try to improve their algorithm by amending the Permutation Key generation while keeping the rest of the encryption algorithm remains the same. The basic idea is that we generate the Permutation Key with some partial information of the plain image where these partial information can also be computed from the ciphertext as well. Alternatively it could be understood as having a function $G : I \rightarrow R$ such $G(I) = G(\pi(I))$ and $G(I)$ is used to generated the Permutation Key. Some statistical functions like mean, standard deviation, max/min, mode, etc satisfies the equality. In the following context, we assume G is chosen as the mean function divided by 256 (the color depth of the image) so that we can assert any pixel change will after the value computed from G while the value of G is normalized between [0, 1].

Permutation Key Generation:

1. Inputs: The encryption key $(\overline{x}_0, p, \hat{x}_0)$, the plain image I (alternatively the scrambled image obtained in the decryption process), and the system parameters (p, q).

2. Outputs: The permutation key (H, L) where H is size m, L with size n.

3. Procedures - Generating the Permutation Key:

(a) Compute $t = \mathrm{G}(I)$ where $G(I) = \dfrac{\sum_{y=1}^{m}\sum_{x=1}^{n} I[y][x]}{m \times n \times 256}$.

(b) The process computes two chaotic sequences $\{\overline{x}_i\}_{i=0}^{\infty}$ *and* $\{\hat{x}_i\}_{i=0}^{\infty}$ using the following iterations with the inputted initial values $\overline{x}_0 \times t$ *and* $\hat{x}_0 \times t$.

$$x_i = 8x_{i-1}^4 - 8x_{i-1}^2 + 1. \quad \textbf{(9)}$$

The result of the process remains the same, namely:

(c) It defines two new sequences $\{\overline{x}_i'\}_{i=1}^{m+n} = \{\overline{x}_i'\}_{i=p+1}^{m+n+p}$ and $\{\hat{x}_i\}_{i=1}^{m+n} = \{\hat{x}_i\}_{i=q+1}^{m+n+q}$. Then, dividing the sequence $\{\overline{x}_i'\}_{i=1}^{m+n}$ into two parts $P_1 = \{\overline{x}_i'\}_{i=1}^{m}$ *and* $P_2 = \{\overline{x}_i'\}_{i=m+1}^{m+n}$. Similarly, the sequence $\{\hat{x}_i\}_{i=1}^{m+n}$ is divided into $Q_1 = \{\hat{x}_i'\}_{i=1}^{n}$ and $Q_2 = \{\hat{x}_i'\}_{i=n+1}^{m+n}$.

(d) P_2 and Q_2 are then sorted ascendingly and the original position of each terms, becomes the sequences S_1 and S_2. So for example if P_2 = {0.4, –0.2, 0.3, 0.5}, S_1 would be {2, 3, 1, 4}. Note the length of S_1 and S_2 would be n and m respectively. Each element in S_1 and S_2 would be a unique integer in the range [1, n] and [1, m] respectively. Then, it reorders the sequences Q_1 and P_1 according S_1 and S_2 respectively, the resulted sequences are called Q_1' *and* P_1'.

(e) Two sequences Q_1' *and* P_1' are then sorted again ascendingly and the original position of each terms becomes the sequences H and L respectively. Note that H has m unique elements from [1, m] and L has n unique elements from [1, n].

The output Permutation key will be used in the Permutation phase of the encryption and the decryption. Note that the decryption would require the scrambled image, which is computed at the inverse of the diffusion process, as an input to generate the permutation key. Since the permutation does not change the statistics of the image thus the selected function G would satisfies $G(\mathrm{I}) = G(\pi(\mathrm{I}))$.

6 Implementation and Analysis

We have implemented Wang et al.'s algorithm and our improvement with C++ on a Windows 7 32-bits desktop machine, running against some sampled images. The running time our algorithm is around 200ms for a 256 × 256 gray image and 700ms for a 720 ×576 gray image. It incurs on average 5.8% additional running time over Wang et al's algorithm. We analysis our algorithm with the following security perspectives to inspect if the encrypted image is statistically random and robust against differential attacks.

6.1 Histogram Analysis

The histogram of a plain image shows the statistically distribution of the pixels. Figure 5, Figure 6, Figure 7 show three sets of the histogram of a plain image and the corresponding cipher image. As we can see the statistical information has been destroyed after encryption.

6.2 Correlation Analysis

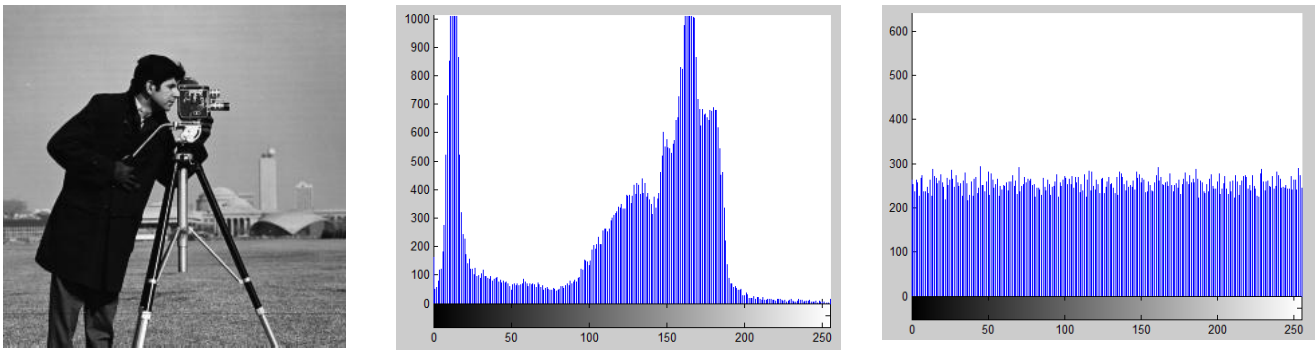
We then look at the correlation between two adjacent pixels. In plain image two adjacent pixels are highly correlated. The correlation coefficients $r_{x,y}$ are defined among pixels by

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad \textbf{(10)}$$

where

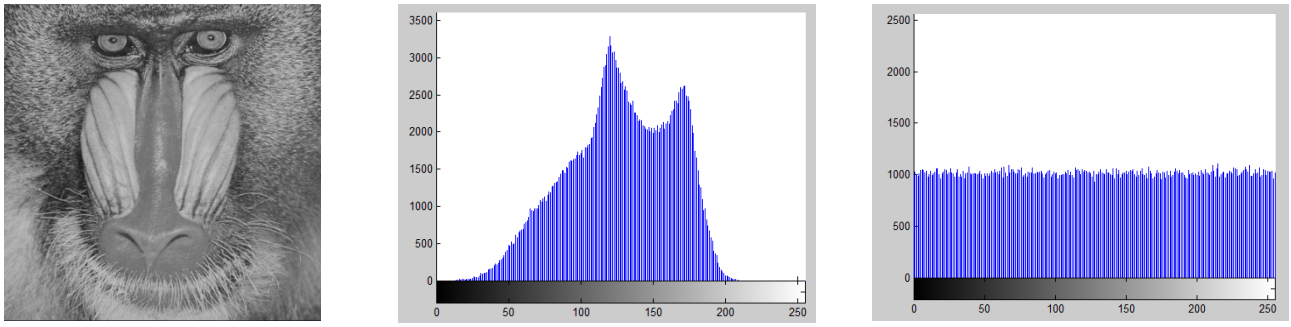
$$cov(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \quad \textbf{(11)}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \textbf{(12)}$$



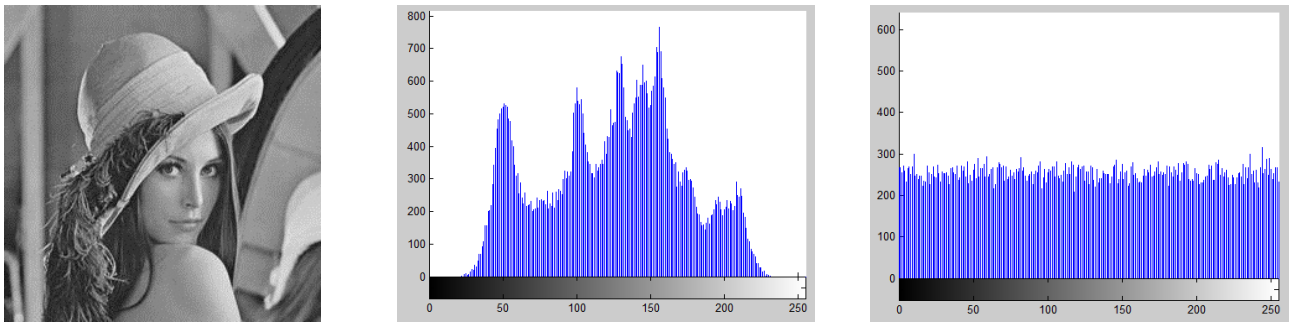
(a) Plain image (b) Histogram of the image (c) Histogram of the Cipher

Figure 5. Histogram analysis of CameraMan



(a) Plain image (b) Histogram of the image (c) Histogram of the Cipher

Figure 6. Histogram analysis of Baboon



(a) Plain image (b) Histogram of the image (c) Histogram of the Cipher

Figure 7. Histogram analysis of Lena

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \quad \textbf{(13)}$$

Here x and y denote the gray values of two adjacent pixels in an image I, while N denotes the number of sampled pixel pairs. The correlation coefficients of two adjacent pixel should be closer to zero to indicate the pixels are less correlated.

We randomly sample the adjacent pixels from a ciphertext in horizontal, vertical, and diagonal directions. Tests are conducted on two sets of 10 images encrypted by Wang et al's algorithm and ours. The result are displayed in Table 3.

With these data we conduct a paired t-Test with confidence level 95% assumption the initial hypothesis H_0 they have a different same mean and alternative hypothesis H_1 is they have the same mean. With $\alpha = 0.05$, $df = 29$, the critical value is given in the t-table $t_{0.05} = 2.045$. The calculated test statistic $t^* = 0.117\,340 < t_{0.05}$ and therefore we reject H_0, i.e., they have the same mean. This conclude that our improvement share the same correlation test performance as Wang et al's algorithm.

Table 3. Result of Correlation Test

Images	Wang et al's algorithm			Our Improvement		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena(256×256)	-0.064	0.0064	0.0147	0.007	0.0139	-0.0063
CameraMan(256×256)	0.0028	-0.0365	0.0423	-0.0353	-0.0209	-0.0047
Baboon (512×512)	-0.0289	0.0568	0.0137	-0.0236	0.0183	0.0025
Boats(720×576)	-0.0272	0.0751	-0.0026	-0.0288	-0.0362	-0.0315
Barbara(720×580)	0.0719	-0.0227	0.0157	0.0172	0.0013	-0.0126
Dog(290×180)	-0.0238	0.0237	-0.0714	-0.0508	0.021	0.0034
Flower(453×502)	-0.0048	-0.0049	-0.0521	0.0833	0.0247	-0.0158
Fruit(444×336)	-0.0137	0.0117	-0.0258	0.0539	0.0047	-0.0084
Girl(244×202)	-0.016	0.0335	0.024	-0.0113	-0.0454	0.0222
Tree(402×265)	-0.0035	0.0265	0.0019	0.0161	0.0052	0.0304

6.3 *NPCR* and *UACI*

NPCR (number of pixels change rate) and *UACI* (Unified average changing intensity) are widely used to evaluate the sensitivity of an image encryption algorithm [2, 4-6, 9-11, 14-15]. They are defined as follows:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{m \times n} \times 100\% \quad \textbf{(14)}$$

and

$$UACI = \left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255 \times m \times n} \right] \times 100\% \quad \textbf{(15)}$$

where

$$D(i,j) = \begin{cases} 0 & C_1(i,j) = C_2(i,j) \\ 1 & C_1(i,j) \neq C_2(i,j) \end{cases} \quad \textbf{(16)}$$

C_1 and C_2 are two cipher images with size $m \times n$ whose the corresponding original plain images have only one pixel different. The ideal value of *NPCR* is254/255 = 0.996 while the ideal value of *UACI* is 0.5 if the statistically distribution of the ciphertexts is random white noise.

The performances of our algorithm against Wang et al's algorithm are summerized in Table 4. It is easy to see that our improvement has a better statistical performance than Wang et al's algorithm in *NPCR*, and *UACI*. This meets our expectation where our improvement is designed to prevent differential attacks.

Table 4. Comparisons of performances between Wang et al.'s algorithm and our improvement

Test	Cipher image (Wang et al.)	Cipher image (Our algorithm)	Improvement(%)
NPCR(%)			
Lena	0.581985	0.996231	71.17%
CameraMan	0.581985	0.995956	71.13%
Baboon	0.741348	0.99601	24.35%
UACI(%)			
Lena	0.195764	0.333317	70.26%
CameraMan	0.196763	0.335368	70.44%
Baboon	0.248687	0.334766	34.61%

6.4 Security Against Differential Attacks

Apparently the differential attack for Wang et al.'s algorithm cannot be directly applied here since the differential images have a different average. Despite this difference is very subtle ($1/(256 \times m \times n)$), Chebyshev's iteration is very sensitive to this difference and the permutation key would therefore be completely different.

Readers may find some sort of modification can be done and mount the same attack against our algorithm. For example, one may consider swapping two pixels in a plain image and analyzing the differential results. In this case, truly the permutation could be retrieved. However, that is only confined to the images with the same pixels-mean. When a challenge cipher is given to an attacker, the value of the pixels-mean is not known to him. Therefore even if the attacker has a set of permutation keys for several different pixels-mean he will be unable to figure out which set of permutation key to try.

7 Conclusion

In this paper we present the vulnerability of Wang et al.'s algorithm and present our improvement with significant performance. The main reason that allows us to break their algorithm is the fact that Permutation

process is deterministic and can be removed by a finite number of oracle queries. Our improvement allows the Permutation process reacts differently against each image. We also recognize the risk of using mean to implement *G*. In our future research we will study how *G* can be better chosen so that it provides a better security statistically and not scarifying too many computation advantage.

Acknowledgements

The authors would thank the reviewers' constructive suggestions and comments. The work of Tsu-Yang Wu was supported in part by the Science and Technology Development Center, Ministry of Education, China under Grant No. 2017A13025 and the Natural Science Foundation of Fujian Province under Grant No. 2018J01636. The work of Chien-Ming Chen was supported in part by Shenzhen Technical Project under Grant number JCYJ20170307151750788 and in part by Shenzhen Technical Project under Grant number KQJSCX20170327161755.

References

[1] T. Gao, Z. Chen, A New Image Encryption Algorithm Based on Hyper-chaos, *Physics Letters A*, Vol. 372, No. 4, pp. 394-400, January, 2008.

[2] S. Behnia, A. Akhshani, H. Mahmodi, A. Akhavan, A Novel Algorithm for Image Encryption Based on Mixture of Chaotic Maps, *Chaos, Solitons & Fractals*, Vol. 35, No. 2, pp. 408-419, January, 2008.

[3] R. Rhouma, S. Belghith, Cryptanalysis of a New Image Encryption Algorithm Based on Hyper-chaos, *Physics Letters A*, Vol. 372, No. 38, pp. 5973-5978, September, 2008.

[4] H. Liu, X. Wang, Color Image Encryption Based on One-time Keys and Robust Chaotic Maps, *Computers & Mathematics with Applications*, Vol. 59, No. 10, pp. 3320-3327, May, 2010.

[5] I. Hussain, T. Shah, M. A. Gondal, An Efficient Image Encryption Algorithm Based on S8 S-box Transformation and NCA Map, *Optics Communications*, Vol. 285, No. 24, pp. 4887-4890, November, 2012.

[6] X. Huang, Image Encryption Algorithm Using Chaotic Chebyshev Generator, *Nonlinear Dynamics*, Vol. 67, No. 4, pp. 2411-2417, March, 2012.

[7] Y. Zhang, D. Xiao, Cryptanalysis of S-box-only Chaotic Image Ciphers against Chosen Plaintext Attack, *Nonlinear Dynamics*, Vol. 72, No. 4, pp. 751-756, June, 2013.

[8] X. Wang, D. Luan, X. Bao, Cryptanalysis of an Image Encryption Algorithm Using Chebyshev Generator, *Digital Signal Processing*, Vol. 25, pp. 244-247, February, 2014.

[9] X. Wang, K. Guo, A New Image Alternate Encryption Algorithm Based on Chaotic Map, *Nonlinear Dynamics*, Vol. 76, No. 4, pp. 1943-1950, June, 2014.

[10] J. Zhao, S. Wang, Y. Chang, X. Li, A Novel Image Encryption Scheme Based on an Improper Fractional-order Chaotic System, *Nonlinear Dynamics*, Vol. 80, No. 4, pp. 1721-1729, June, 2015.

[11] F.-G. Jeng, W.-L. Huang, T.-H. Chen, Cryptanalysis and Improvement of Two Hyper-chaos-based Image Encryption Schemes, *Signal Processing: Image Communication*, Vol. 34, pp. 45-51, May, 2015.

[12] W.-S. Yap, R. C.-W. Phan, W.-C. Yau, S.-H. Heng, Cryptanalysis of a New Image Alternate Encryption Algorithm Based on Chaotic Map, *Nonlinear Dynamics*, Vol. 80, No. 3, pp. 1483-1491, May, 2015.

[13] X. Wang, L. Liu, Y. Zhang, A Novel Chaotic Block Image Encryption Algorithm Based on Dynamic Random Growth Technique, *Optics and Lasers in Engineering*, Vol. 66, pp. 10-18, March, 2015.

[14] X.-Y. Wang, Y.-Q. Zhang, X.-M. Bao, A Novel Chaotic Image Encryption Scheme Using DNA Sequence Operations, *Optics and Lasers in Engineering*, Vol. 73, pp. 53-61, October, 2015.

[15] L. Xu, Z. Li, J. Li, W. Hua, A Novel Bit-level Image Encryption Algorithm Based on Chaotic Maps, *Optics and Lasers in Engineering*, Vol. 78, No. 4, pp. 17-25, March, 2016.

[16] G. Ye, X. Huang, A Feedback Chaotic Image Encryption Scheme Based on Both Bit-level and Pixel-level, *Journal of Vibration and Control*, Vol. 22, No. 5, pp. 1171-1180, March, 2016.

[17] Q. Zhang, Y. Guo, W. Li, Q. Ding, Image Encryption Method Based on Discrete Lorenz Chaotic Sequences, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 7, No. 3, pp. 576-586, May, 2016.

[18] Y. Suryanto, Suryadi, K. Ramli, A Secure and Robust Image Encryption Based on Chaotic Permutation Multiple Circular Shrinking and Expanding, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 7, No. 4, pp. 697-713, July, 2016.

[19] X. Liao, F. Chen, K.-W. Wong, On the Security of Public-key Algorithms Based on Chebyshev Polynomials over the Finite Field ZN, *IEEE Transactions on Computers*, Vol. 59, No. 10, pp. 1392-1401, October, 2010.

[20] C.-M. Chen, L. Xu, T.-Y. Wu, C.-R. Li, On the Security of a Chaotic Maps Based Three-party Authenticated Key Agreement Protocol, *Journal of Network Intelligence*, Vol. 1, No. 2, pp. 61-66, May, 2016.

[21] C.-M. Chen, W. Fang, K.-H. Wang, T.-Y. Wu, Comments on an Improved Secure and Efficient Password and Chaos-based Two-party Key Agreement Protocol, *Nonlinear Dynamics*, Vol. 87, No. 3, pp. 2073-2075, February, 2017.

[22] C.-M. Chen, C.-T. Li, S. Liu, T.-Y. Wu, J.-S. Pan, A Provable Secure Private Data Delegation Scheme for Mountaineering Events in Emergency System, *IEEE Access*, Vol. 5, No. 1, pp. 3410-3422, February, 2017.

[23] M. Yu, Z. Du, X. Liu, D. Qun, H. Chen, The Method of Obtaining Best Unary Polynomial for the Chaotic Sequence of Image Encryption, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 8, No. 5, pp. 1103-1110,

September, 2017.
[24] H. Zhu, Y. Zhang, An Efficient Chaotic Maps-based Deniable Authentication Group Key Agreement Protocol, *Wireless Personal Communications*, Vol. 96, No. 1, pp. 217-229, September, 2017.
[25] N. Lin, H. Zhu, Enhancing the Security of Chaotic Maps-based Password-authenticated Key Agreement Using Smart Card, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 8, No. 6, pp. 1273-1282, November, 2017.
[26] C.-M. Chen, K.-H. Wang, T.-Y. Wu, E. K. Wang, On the Security of a Three-party Authenticated Key Agreement Protocol Based on Chaotic Maps, *Data Science and Pattern Recognition*, Vol. 1, No. 2, pp. 1-10, December, 2017.
[27] D. Fang, S. Sun, A New Scheme for Image Steganography Based on Hyperchaotic Map and DNA Sequence, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 9, No. 2, pp. 392-399, March, 2018.
[28] L. Wang, Q. Ye, Y. Xiao, Y. Zou, B. Zhang, An Image Encryption Scheme Based on cross Chaotic Map, *CISP 2008-2008 International Congress on Image and Signal Processing*, Sanya, China, 2008, pp. 22-26.
[29] M. I. Sobhy, A.-E. Shehata, Methods of Attacking Chaotic Encryption and Countermeasures, *ICASSP 2001 - 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake, UT, 2001, pp. 1001-1004.
[30] I. Hussain, T. Shah, H. Mahmood, A New Algorithm to Construct Secure Keys for AES, *International Journal of Contemporary Mathematical Sciences*, Vol. 5, No. 26, pp. 1263-1270, January, 2010.
[31] C.-M. Chen, B. Xiang, K.-H. Wang, K.-H. Yeh, T.-Y. Wu, A Robust Mutual Authentication with a Key Agreement Scheme for Session Initiation Protocol, *Applied Sciences*, Vol. 8, No. 10, 1789, October 2018.
[32] K.-H. Wang, C.-M. Chen, W. Fang, T.-Y. Wu, On the Security of a New Ultra-lightweight Authentication Protocol in IoT Environment for RFID Tags, *Journal of Supercomputing*, Vol. 74, No.1, pp. 65-70, January, 2018.
[33] K.-H. Yeh, C. Su, J.-L. Hou, W. Chiu, C.-M. Chen, A Robust Mobile Payment Scheme with Smart Contract-based Transaction Repository, *IEEE Access*, Vol. 6, pp. 59394-59404, October 2018.

Biographies

Tsu-Yang Wu is currently an Associate Professor in College of Computer Science and Engineering at Shandong University of Science and Technology. He serves as executive editor in Journal of Network Intelligence and as associate editor in Data Science and Pattern Recognition. His research interests include cryptography and Network security.

Xiaoning Fan received the M.S. degree in School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. Currently, she is an engineer in Huawei, China. Her research interest includes information security.

King-Hang Wang received his Ph.D. from the National Tsing Hua University. He worked in the Hong Kong Institute of Technology in 2010 as a lecturer. He joined the Hong Kong University of Science and Technology since 2015. His research focus is cryptography, mobile security, and provable authentication.

Jeng-Shyang Pan is currently the Dean in College of Information Science and Engineering and an Assistant President at Fujian University of Technology, China. He is the IET Fellow, UK and was offered Thousand Talent Program in China in 2010. His research interests include information security and artificial intelligence.

Chien-Ming Chen is currently an Associate Professor in School of Computer Science and Technology at Harbin Institute of Technology (Shenzhen). He serves as an Associate Editor in Journal of Information Hiding and Multimedia Signal Processing, Data Science and Recognition, and Journal of Network Intelligence. His research interests include network security, and cryptography.