Image Encryption Using Partitioned Cellular Automata

Yong Wang[1,2] Yi Zhao[2] Qing Zhou[3] Zehui Lin[1]

1 Key Laboratory of Electronic Commerce and Logistics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

2 College of Computer Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

3 College of Computer Science, Chongqing University, Chongqing 400044, China

Abstract: Image encryption techniques aims to protect the content of image with higher efficiency and security than conventional cryptographic methods by making use of special properties of image. This paper presents an image encryption model based on two-dimensional partitioned cellular automaton. The model has the same topology as a digital image and is flexible to images with different color depth; it is efficient as only substitution and permutation operations are involved; the properties of cellular automata make the model easy for Very Large Scale Integration(VLSI) implementation. Moreover, unlike most known image encryption algorithms, this model can support parallel computing. A probability cellular automaton called coloring model, is proposed to study the sensitivity of the encryption model. It shows that the model meets the global strict avalanche criterion in at most $M+N+7$ rounds of encryption for an $M \times N$ image. Several approaches are proposed to estimate the minimal number of rounds to fulfill the global strict avalanche criterion; simulation shows that the maximal error is only one round. An image encryption algorithm based on this model is presented, which is demonstrated by experiments to own the properties of randomness and sensitivity.

Keywords: **partitioned cellular automata; image encryption; strict avalanche criterion; parallel encryption;**

1. Introduction

The past decade has witnessed significant advancement in multimedia processing techniques and communication techniques [1-5]. Due to the commercial success in digital cameras and 3G phones, along with the increasing popularity of various Web 2.0&3.0 applications and social networking websites, the digital images are being created, distributed, and stored at a high speed that has never been experienced before. Accordingly, confidentiality becomes an important issue for secure distributions and storages of digital images, especially for those involving business values or personal privacies.

Cryptographic techniques play a crucial role in the protection of the digital image contents from unauthorized eavesdroppers. Naïve approaches which use conventional encryption standards such as International Data Encryption Algorithm(IDEA), Advanced Encryption Standard(AES) to encrypt image data, are found to be insecure in some circumstances [6]. Moreover, the way that the naïve approaches encrypt digital images limits their processing speed, as they treats image data only as a structure-less binary sequence, taking no advantage of the properties of digital image such as visibility, compressibility and extensity. In contrast, image encryption techniques aim to achieve better security and efficiency by making use of properties of digital images.

Image encryption techniques can be divided into two categories. The first one integrates encryption algorithm with some image compression technique, such as [7-9], trying to compromise between

compression ratio and security level. The second category of image encryption technique, also known as the space-domain image encryption technique, carries out the encryption directly on the uncompressed image data. The space-domain image encryption technique is preferable for applications where the encryption efficiency is of great concern, since compression is usually much slower than encryption [10]. We focus on the second category of image encryption technique in the following discussion.

Of all space-domain image encryption techniques, chaos-based image encryption algorithms are most widely studied. Chaotic systems are famous for being extremely sensitive to both initial conditions and system parameters. They also exhibit other characteristics such as pseudo-randomness and ergodicity. Besides, two-dimensional chaotic map works well on image as a permutation operation. For example, Baker map [11], Cat map [12] or Standard map [13] is reported to be helpful in achieving high security as a critical component in the image encryption algorithm. Nevertheless, chaotic maps require a large amount of multiplication and division operations, which reduce the encryption speed drastically. A few space-domain image encryption algorithms replace the chaotic map with other operations. For example, Bourbakis etc. use the 'SCAN', a formal language originally designed for pattern recognition, to permute the pixels [14].

Almost all the space-domain image encryption algorithms adopt an encryption mode similar to ciphertext block chain (CBC) mode [15]. This mode helps the image encryption algorithms to possess the property of "global sensitivity", i.e., change in one bit of the plain-image resulting in change of the whole cipher-image. Algorithms using this mode, however, are inefficient when running on a parallel computing platform. Support of parallel computing is of great importance for image and video processing [16,17]. For example, a parallel image encryption framework is proposed in [18], in which it was found that two dimensional cellular automaton (CA) is an appropriate model for parallel image encryption.

CA is a dynamic system discretized in both space and time. It consists of a regular grid of cells. The next state of each cell is determined by the current states of its neighbors and itself. CA is famous for its simple, regular, parallel and locally interconnected characteristics.

Cellular automata find their wide applications in conventional cryptography areas. Wolfram designed a stream cipher with a nonlinear CA rule [19]. The use of CA in block cipher [20] and public key cryptography [21] were also reported. Nevertheless, CA is mostly applied to the design of pseudorandom number/pattern generator, such as in [22-23], due to its excellent pseudo-randomness property.

Some image encryption algorithms have been proposed based on CA. Most of them use CA as a pseudorandom generator; A few algorithms adopt second-order CA or its variants. In this paper, we propose an image encryption model using two dimensional partitioned cellular automata (PCA). A two dimensional PCA comprises of a regular grid of cells; each cell can be further divided into a fixed number of parts. The state of each cell is affected by two functions: function p moves each part of a cell to one of its neighbors; function f changes the state of each part. When used for image encryption, each cell stands for a pixel, and each part of a cell consists of one or more bits of a pixel. The states of all cells, which are initiated by the plain-image, turn to the cipher-image after sufficient number of rounds of evolutions of the PCA.

The use of second-order CA or its variants for image encryption has some problems: the ciphertext is usually larger than the plaintext; the algorithm cannot support parallel computing efficiently; or it cannot achieve the "global sensitivity". Image encryption using PCA does not have these problems. The number of cells remains unchanged before and after the evolution of PCA; all the cells change to

their new states simultaneously; and the influence of each cell spreads out to its surrounding cells due to the joint effect of functions *p* and *f* in several rounds of evolutions. Moreover, PCA preserves all the merits of CA, which makes the implementation of the image encryption algorithm fast, easy and cheap. Finally, unlike most known image encryption techniques, the PCA-based model is flexible to digital images with various color depth.

We summarize our contributions as follows.

- We propose an image encryption mode based on the PCA. The model is reversible, efficient and can support parallel computing; besides, it is especially suitable for VLSI implementation. We investigate the basic requirements for functions *p* and *f* from the point of view of security. We also show the relationship between the two functions and the conventional cryptographic methods.
- We analyze the sensitivity of the encryption model during its evolutions and give a theoretic upper bound of the number of rounds to fulfill the global strict avalanche criterion (global SAC), a stringent measurement for "global sensitivity". We propose a probabilistic CA model called coloring model to track how the change of one bit in the plain-image propagates to all pixels of the cipher-image.
- Based on our proposed image encryption model, we introduce an image encryption algorithm, which possesses excellent properties of randomness and sensitivity.

The remainder of this paper is organized as follows: some related work and fundamental concepts are presented in Section 2 and 3, respectively. In section 4, we introduce our image encryption mode based on PCA. In section 5, we investigate the sensitivity of the encryption model and propose several approaches to estimate the minimal number of rounds to satisfy the global SAC. Finally, in sections 6 and 7 we present a practical image encryption algorithm and validate its performance through extensive simulation.

2. Prior Work

The use of CA for image encryption has been suggested by researchers. For example, some algorithms use one or two dimensional CAs to generate pseudorandom numbers, then each pixel of the image is encrypted through the XOR or substitution operation with a pseudorandom number [24-26]. The security of these algorithms relies on the randomness of these numbers. Hence, they belong to the category of stream ciphers.

Alvarez *et al.* proposed an image encryption algorithm based on the second-order CA [27], where the next state of a cell is determined not only by the current states of its neighbors but also by its previous state. In order to recover the initial state of a second-order CA, one must store the last two states of each cell. Therefore, the cipher-image is two times larger than the plain-image.

Maleki *et al.*, prevent the expansion of cipher-image by extending the second-order CA to a higher-order one at the expense of image quality [28], as the least significant bit-plane (LSB) of the image cannot be recovered after decryption.

Yu *et al.* proposed an innovative image encryption algorithm based on the symmetrical-coupled toggle CA [29], a variant of second-order CA, where cells are divided into two groups. The next state of the cell in each group depends on the current states of cells in both groups. The symmetrical-coupled toggle CA avoids the loss of image quality and the expansion of cipher-image, however, each pixel is encrypted independently and there is no interaction between pixels. Therefore, no property of "global sensitivity" is presented.

3. Preliminaries

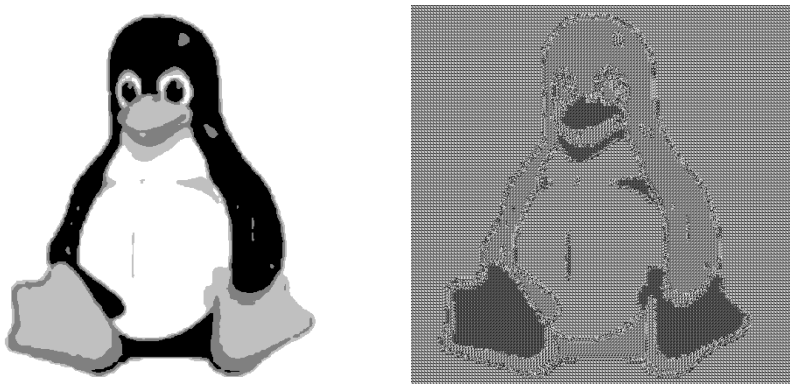
在这一部分中，我们将介绍与基于PCA的图像加密模型有关的一些基本概念，以及一些设计和评估图像加密算法的准则。

In this section, we introduce some fundamental concepts pertaining to PCA-based image encryption model, and some criterions for design and evaluation of image encryption algorithms.

3.1 Framework of Multiprocessor-Collaboration Encryption

多处理器协同加密框架

A cipher-image generated by naïve approaches may leak information about the plain-image. Fig. 1 (b) shows the cipher-image derived by Advanced Encryption Standard with electronic codebook (ECB) mode corresponding to the plain-image depicted in Fig. 1 (a).The disadvantage of ECB mode is that blocks are encrypted independently of other blocks [15]. So ECB mode of encryption possesses no property of "global sensitivity". In fact, Fig. 1 (b) still reveals the content of the plain-image.



(a) (b)

Fig. 1. Image Encryption by AES. (a) plain-cipher (b) cipher-image

In order to overcome this problem, most space-domain image encryption algorithms adopt a mode similar to CBC [11-14], which at a high level, can be expressed as:

$$c(i) = c(i-1) \oplus E\big(p(i)\big) \tag{1}$$

where $p(i)$ and $c(i)$ are the ith pixel of the plain-image and cipher-image respectively, and $E(.)$ denotes an encryption operation. Formula (1) indicates that the change to one pixel causes change to all its following pixels. But this mode is computationally inefficient as encryption of current pixel cannot start until that of the previous pixel ends, hence, only one pixel is encrypted at a time even in a platform supporting parallel computing.

To circumvent this problem, a framework of multiprocessor-collaboration encryption has been proposed [18], which preserves the property of 'global sensitivity' but also supports parallel computing. In this framework, each processor has its own memory and can encrypt a number of pixels simultaneously; encryption algorithms under this framework maintain 'global sensitivity' property through communications between processors. In this work, we demonstrate that PCA is the most suitable approach for implementation of this framework.

3.2 Partitioned Cellular Automata

分区的元胞自动机

A k-dimensional m-neighbor partitioned cellular automata (PCA) is a system defined by [30]

$$P = (\mathbb{z}^k, Q, S, f, \#) \tag{2}$$

where $\mathbb{z}$ is the set of all integers, Q is a set of states of each cell, $S = \{s_1, s_2, \ldots, s_m\}$ is a set of neighborhoods and $s_i \in \mathbb{z}^k$ $(i = 1, \ldots, m)$, $f: Q \to Q$ is the PCA rule, and $\# \in Q$ is a quiescent state satisfying $f(\#) = \#$; each cell is further divided into m parts, and $Q_i (i = 1, \ldots, m)$ is a non-empty finite set of states of the i-th part of each cell (thus $Q = Q_1 \times \ldots \times Q_m$). If the same PCA rule applies to all cells, the PCA is called uniform; otherwise, the PCA is called hybrid.

均匀

混合

The evolution of a PCA is determined by (3): [30]

$$\alpha^{(r+1)}(x) = f\left(p_1\left(\alpha^{(r)}(x+s_1)\right), \ldots, p_m\left(\alpha^{(r)}(x+s_m)\right)\right) \tag{3}$$

where $\alpha^{(r)}(x)$, the global configuration of the cellular automata in r-th round, is a mapping $\alpha^{(r+1)}: \boldsymbol{Z}^k \to Q$, and $\alpha^{(r)}(x)$ represents the state of the cell on the coordinate x after r rounds of iterations of f. p_i is a projection function such that

$$p_i(x) = x \cdot e_i \tag{4}$$

and symbol · denotes the dot product; e_i is a standard basis of m-dimensional space, i.e., $e_i = [a_1, a_2, \ldots, a_m]$, $a_i = 1$, $a_j = 0$, $j \neq i$. The effect of p_i is like that a cell moves the ith part of its ith neighbor to the ith part of itself.

The cyclic (or periodic) and null (or fixed) boundary conditions are two types of boundary conditions usually considered for finite PCA [20]. In the case of the cyclic boundary conditions, the cells at the beginning and the end of each row (or each column) are considered adjacent, resulting in a circular grid for the one-dimensional case and a toroidal one for the two dimensional case. In the case of the null boundary conditions, the grid is surrounded by an outer layer of cells in the fixed state of zeros.

Two-dimensional PCA, where the cells are arranged in rows and columns, has two typical neighborhoods: the Von Neumann neighborhood and Moore neighborhood. Von Neumann neighborhood consists of four orthogonally surrounding neighbors, and Moore neighborhood comprises four Von Neumann neighborhoods and four diagonally surrounding neighbors.

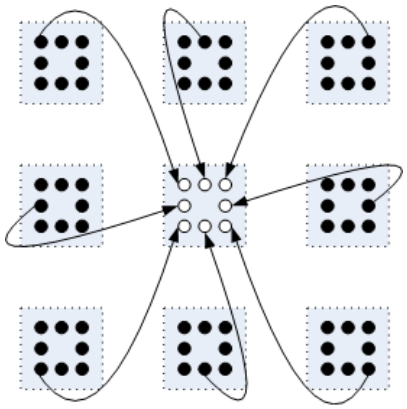


Fig. 2. An Example of PCA with the Moore Neighborhoods

Fig. 2 depicts an example of a two-dimensional PCA with Moore neighborhoods, where the values of the neighborhoods are assigned as (5)

$s_1 = (1, -1)$, $s_2 = (1, 0)$, $s_3 = (1, 1)$, $s_4 = (0,-1)$, $s_5 = (0, 1)$, $s_6 = (-1,-1)$, $s_7 = (-1, 0)$, $s_8 = (-1, 1)$, and

$$s_t = (s_{t,1}, s_{t,2}), \ t = 1, \ldots, 8 \tag{5}$$

3.3 Some Fundamental Concepts for Image Encryption Techniques

图像加密技术的几个基本概念

Confusion and diffusion, identified by Claude Shannon[31], are two methods to design block ciphers as well as image encryption algorithms. Confusion complicates the relationship between the key and ciphertext; diffusion spreads out the local statistical structure of the plaintext over the ciphertext. A

well-designed cipher possesses randomness and sensitivity properties by methods of confusion and diffusion.

- Randomness property: Each bit of the ciphertext is equally likely to be 0 and 1; moreover, any bits of the ciphertext are statistically independent of the key and plaintext.
- Sensitivity property: A slight variation of the plaintext results in significant change of the ciphertext. In particular, we use SAC to measure the sensitivity of a Boolean function.

A Boolean function $f(x)$ maps an element of $\{0,1\}^n$ to an element of $\{0,1\}$. A balanced Boolean function outputs the same number of zeros to that of ones over its input sets. The Hamming weight hw of an element of $\{0,1\}^n$ equals the number of ones in that element, i.e.

$$hw(x) = \sum_{i=1}^{n} x_i \tag{6}$$

where $x = (x_1, x_2, ..., x_n)$, $x_i \in \{0,1\}$.

Definition 3.1. Strict Avalanche Criterion (SAC): A Boolean function $f(x)$ satisfies SAC if and only if its output changes with a probability of one half whenever a single bit of x is flipped [32], i.e., for all $hw(a)=1$,

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x) \oplus f(x \oplus a) = \frac{1}{2} \tag{7}$$

where $a \in \{0,1\}^n$, and $\oplus$ denotes bit-wise exclusive-OR operation.

Similarly, we use the global SAC as a measurement of the "global sensitivity" in image encryption techniques.

Definition 3.2. Global SAC: An encryption algorithm satisfies the global SAC if, for each key, each bit of the cipher-image changes with a probability of one half whenever a single bit of the plain-image is changed.

The concept of SAC is generalized to a stronger cryptographic criterion by Meier and Staffelbach[33].

Definition 3.3. Perfect Nonlinear Boolean Function: A Boolean function $f(x)$ is called perfect nonlinear if and only if (3) holds for all $1 \le hw(a) \le n$ [33]

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x) \oplus f(x \oplus a) = \frac{1}{2} \tag{8}$$

A perfect nonlinear Boolean function may not fulfill other cryptographic requirements. In that case, we only require the function be near-perfect nonlinear.

Definition 3.4. Near-perfect Nonlinear Boolean Function: A Boolean function $f(x)$ is called near-perfect nonlinear if and only if $r(a)$ is close to one half for all $1 \le hw(a) \le n$:

$$r(a) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x) \oplus f(x \oplus a) \approx \frac{1}{2} \tag{9}$$

4. An Encryption Model Based on the PCA

一种基于PCA的加密模型

We choose two-dimensional PCA as the encryption model as its topology is most appropriate for digital images. The proposed PCA is uniform and has cyclic boundary conditions, so all cells can share a same operation, which makes the implementation of the PCA easier. The PCA uses the Moore neighborhood, as Von Neumann neighborhood cannot meet the global SAC.

Each cell in the PCA represents a pixel with a precision of $8n$ bits. A cell is divided into 8 parts and

each part contains n bits. Therefore, the encryption model is flexible to encrypt digital images with various color depths by adjusting n. Table 1 lists the relationship between n and types of image with different color depths.

Table 1. The relationship between n and types of images with different color depths

n	Type of image
1	Grayscale image
2	High color image
3	True color image
>3	Deep color image

The neighborhoods of the proposed PCA are assigned with values given by (5). Once the value of each neighborhood is fixed, formula (3) can also be expressed as a composition of functions f and p:

$$f\left(p_1\left(\alpha(x+s_1)\right),\ldots,p_m\left(\alpha(x+s_m)\right)\right)=f\circ p(x) \tag{10}$$

where p stands for all projection operations on the Moore neighborhood of a cell. Function p, which moves different parts between cells, is essentially a cryptographic permutation operation. Since p helps to distribute the statistical characteristics in local area to a wider area after several rounds of evolutions, it belongs to the method of diffusion.

The behavior of a PCA depends mainly on the property of function $f:\{0,1\}^{8n}\to\{0,1\}^{8n}$. f can be represented by $8n$ Boolean functions $f_1,\ldots,f_{8n}$, where $f_j:\{0,1\}^{8n}\to\{0,1\}$, $j=1,\ldots,8n$, and

$$f(x)=(f_1(x),\ldots,f_{8n}(x)) \tag{11}$$

We require that function f possess two properties:

(1) f is reversible;

(2) each f_j (j=1,…, 8n) is near-perfect nonlinear.

Morita has proved that the PCA is reversible if and only if f is reversible [29]; therefore the first property ascertains that the encryption model can be decrypted correctly. The second property helps the proposed image encryption model to satisfy the global SAC. Note that f is reversible implies that each f_j is balanced; otherwise, there are at least two different elements x and x' such that $f(x) = f(x')$ and f becomes irreversible. Meier and Staffelbach proves that a balanced Boolean function can never be perfect nonlinear [33]. Therefore, we require that All f_j are near-perfect nonlinear.

Since f is a reversible function with its domain and the codomain both on $\{0, 1\}^{8n}$, it acts as a cryptographic substitution operation, the primary method of confusion. Function f could be as simple as a single fixed S-box or as complex as some known encryption standards. However f is implemented, we considered it as an abstract substitution operation when analyze the sensitivity of the encryption model.

The proposed encryption model is consistent with the multiple-processor collaboration encryption framework, where each processor encrypts a pixel via function f and the processors communicate via function p.

The encryption model based on PCA has many advantages when applied to image encryption:

(1) Suitability for digital images: The two-dimensional PCA has the same topology as a digital image,

so each pixel can be processed by a corresponding cell in a rectangular grid. Moreover, the encryption model is adaptive to images with different color depths;

(2) High-efficiency in execution: The proposed PCA involves only substitution and permutation operations, which guarantees that image encryption be executed with high efficiency;

(3) Parallelism of computation: PCA evolves with each cell being updated simultaneously and independently of each other, which makes it suitable for parallel computing;

(4) Amenability to VLSI implementation: The uniformity (All cells share a common PCA rule) and the locality (the state of each cell only depends on its neighborhoods) make it easier to implement the encryption model with VLSI circuits.

In addition to above advantages, we show that the proposed encryption model satisfies the global SAC in the following section.

5. Analysis of Sensitivity of the PCA-Based Encryption Model

In this section, we analyze the sensitivity of our proposed image encryption model. We concern following questions: how does the slight difference in plain-images propagate inside the cipher-images during encryption? Can the encryption model satisfy the global SAC? If so, in how many rounds can it satisfy the global SAC? In order to answer these questions, we introduce coloring model, a probability CA corresponds to the sensitivity property of our proposed encryption model.

5.1 Coloring Model

Intuitively, if two plain-images differ in only a single bit, the corresponding cipher-images differ in a couple of pixels after one round evolution of our proposed encryption model. After more number of rounds of evolutions, the difference between two cipher-images is enlarged. Analysis of the sensitivity of the model tells us how the difference spreads over the whole image during encryption.

Let $C_1^{(r)}(i,j)$ and $C_2^{(r)}(i,j)$ denote pixels at *i*th row and *j*th column of two cipher-images which are produced after r rounds of encryptions of two plain-images P_1 and P_2. We use $\alpha^{(r)}(i,j)$ to represent whether they are equal or not:

$$\alpha^{(r)}(i,j) = \begin{cases} 0, & C_1^{(r)}(i,j) \neq C_2^{(r)}(i,j) \\ 1, & C_1^{(r)}(i,j) = C_2^{(r)}(i,j) \end{cases} \tag{12}$$

Fig. 3 shows the relationship between $\alpha^{(r)}$ and two cipher-images $C_1^{(r)}$ and $C_2^{(r)}$. By focusing on the difference of pixels instead of their values, we can investigate the sensitivity of the proposed image encryption model via a simpler CA, which we call coloring model.

The coloring model is a probability CA, which is characterized by following rules:

(1) The coloring model is a two dimensional CA with the cyclic boundary conditions and Moore neighborhood (See Fig. 4);

(2) Each cell has two possible states, black(or 0) and white(or 1);

(3) Initially, all the cells except one are white;

(4) A white cell can only dye its neighbors white; a black cell with some probability can dye its neighbors black. A cell (regardless of its current color) changes its color to black if at least one of its neighbors dyes it black; otherwise, the color of the cell turns to white.

(5) All cells are colored simultaneously.

Let $\alpha^{(r)}(i,j)$ denotes the color of the cell (i, j) after r rounds of evolutions of the coloring model. $\alpha^{(r)}(i,j) = 0$ means that pixels (i, j) of ciphertexts $C_1^{(r)}$ and $C_2^{(r)}$ are different after evolutions of the PCA-based encryption model. Therefore, the coloring model is a simplified version of the proposed encryption model in regard to the difference of the cipher-images. We expect to reach some conclusions

on the sensitivity of the PCA-based encryption model by tracking the expansion of the black cells of the coloring model.

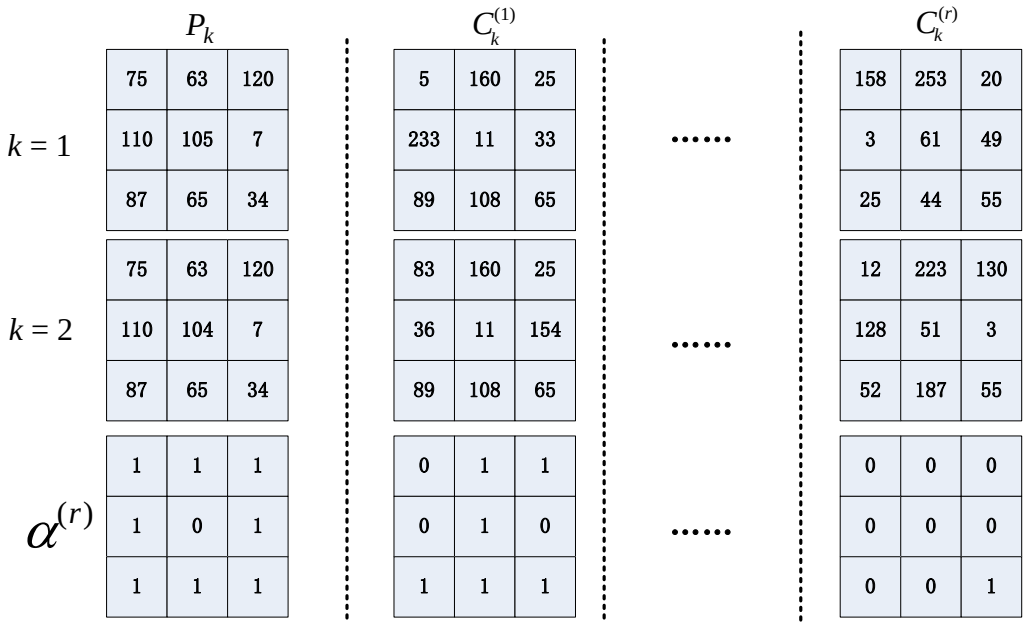


Fig. 3 The difference of pixels of cipher-images

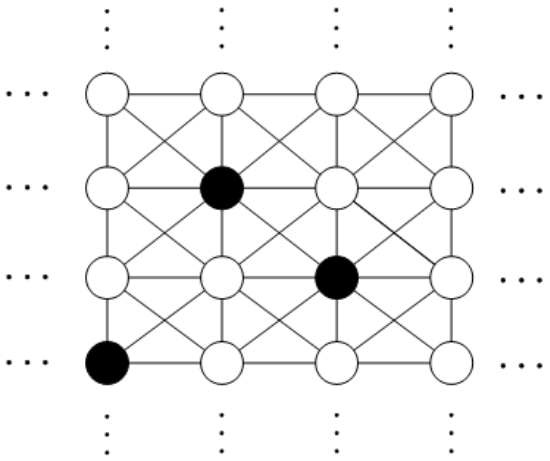


Fig. 4 The Coloring Model

5.2 Evolution of the Coloring Model

Since the coloring model is a uniform CA with cyclic boundary conditions, any cell can be considered as the central one in the model. We assume the coordinate of the unique black cell is (0, 0). We use $h^{(r)}(i,j)$ to denote the probability that cell (i, j) happens to be white after r rounds of evolutions:

$$h^{(r)}(i,j) = \mathrm{P}\{\alpha^{(r)}(i,j) = 1\} \tag{13}$$

From rule (3) of coloring model, the initial probability of each cell is given by

$$h^{(0)}(i,j) = \begin{cases} 0, & i = 0, j = 0 \\ 1, & \text{otherwise} \end{cases}. \tag{14}$$

$h^{(r)}(i,j)$ can be computed recursively. To simplify the discussion, let us assume all Boolean

function corresponding to f are perfect nonlinear, and their outputs are independent of each other. Hence, if a cell changes, after the operation of f, each bit of the cell will change with a probability of 0.5 independently; each part of a cell contains n bits, so each part remains unchanged with a probability of 2^{-n}. After the operation of function p, the tth part of a cell is replaced by a part of its tth neighbor. So a cell will stay unaffected from its tth neighbor with a probability of 2^{-n} if its tth neighbor changes. On the other hand, a cell will not be affected by its tth neighbor given that this neighbor does not change. In the coloring model, this fact is formulated as:

$$\mathrm{P}\left\{\alpha_t^{(r)}(i,j)=1 \middle| \alpha^{(r-1)}\left(i+s_{t,1}, j+s_{t,2}\right)=0\right\}=2^{-n},$$

$$\mathrm{P}\left\{\alpha_t^{(r)}(i,j)=1 \middle| \alpha^{(r-1)}\left(i+s_{t,1}, j+s_{t,2}\right)=1\right\}=1 \tag{15}$$

where $s_{t,1}$ and $s_{t,2}$ are defined in (5), and $\alpha_t^{(r)}(i,j)$ denotes the color of the cell (i, j) dyed by its tth neighbor in rth round. Thus,

$$\begin{aligned}
&\mathrm{P}\left\{\alpha_t^{(r)}(i,j)=1\right\} \\
&= \mathrm{P}\left\{\alpha^{(r-1)}\left(i+s_{t,1}, j+s_{t,2}\right)=0\right\}\mathrm{P}\left\{\alpha_t^{(r)}(i,j)=1 \middle| \alpha^{(r-1)}\left(i+s_{t,1}, j+s_{t,2}\right)=0\right\} \\
&\quad + \mathrm{P}\left\{\alpha^{(r-1)}\left(i+s_{t,1}, j+s_{t,2}\right)=1\right\}\mathrm{P}\left\{\alpha_t^{(r)}(i,j)=1 \middle| \alpha^{(r-1)}\left(i+s_{t,1}, j+s_{t,2}\right)=1\right\} \\
&= \left(1-h^{(r-1)}\left(i+s_{t,1}, j+s_{t,2}\right)\right)2^{-n} + h^{(r-1)}\left(i+s_{t,1}, j+s_{t,2}\right)\times 1 \\
&= (1-2^{-n})h^{(r-1)}\left(i+s_{t,1}, j+s_{t,2}\right) + 2^{-n}
\end{aligned} \tag{16}$$

Assuming the dyeing of each neighborhood is independent of each other, we have

$$h^{(r)}(i,j)=\prod_{t=1}^{8}\mathrm{P}\left\{\alpha_t^{(r)}(i,j)=1\right\} \tag{17}$$

From (14) and (17), for any positive integer r and any integer i, j, we have

$$h^{(r)}(i,j)=h^{(r)}(-i,j)=h^{(r)}(i,-j)=h^{(r)}(-i,-j) \tag{18}$$

which can be proved by induction. So we only need to calculate $h^{(r)}(i,j)$ for the cells in quadrant I ($0\le i\le \lfloor M/2\rfloor, 0\le j\le \lfloor N/2\rfloor$) in the following discussion.

Lemma 1. For $i > -1$, $j > 0$ and $r > 1$,

$$h^{(r)}(i,j-1)\le h^{(r)}(i,j) \tag{19}$$

Proof. . Lemma 1 can be proved by induction.

Table 2 lists all the values of $h^{(2)}(i,j)$ calculated according to (14) ~ (17), where $q = 2^{-n+1}-2^{-2n} < 1$. The difference between each pair of $h^{(2)}(i,j-1)$ and $h^{(2)}(i,j)$ in Table 2 demonstrates that (19) is true for $r = 2$.

Table 2 The values of $h^{(2)}(i,j)$ for $i>-1$, $j>-1$

	$j = 0$	$j = 1$	$j = 2$	$j > 2$
$i > 2$	1	1	1	1
$i = 2$	q^3	q^2	q	1
$i = 1$	q^4	q^2	q^2	1
$i = 0$	q^8	q^4	q^3	1

We assume (19) is true when $r = k-1 > 1$, i.e.,

$$h^{(k-1)}(i,j-1)\le h^{(k-1)}(i,j)$$

From (16), for $1 \le t \le 8$, $t \in \mathbf{Z}$, we have

$$\mathrm{P}\left\{\alpha_t^{(k)}(i, j-1) = 1\right\} \le \mathrm{P}\left\{\alpha_t^{(k)}(i, j) = 1\right\} \tag{20}$$

By (17), we have

$$h^{(k)}(i, j-1) \le h^{(r)}(i, j) \tag{21}$$

Therefore, (19) is true when $r = k$, which, by the Principle of Induction, proves the Lemma 1.

Theorem 1. For $0 \le i \le k$, $0 \le j \le l$ and $r > 1$,

$$h^{(r)}(i, j) \le h^{(r)}(k, l) \tag{22}$$

Proof. Similar to (19), we have

$$h^{(r)}(i-1, j) \le h^{(r)}(i, j) \tag{23}$$

So, $h^{(r)}(i, j) \le h^{(r)}(i+1, j) \le \cdots \le h^{(r)}(k, j) \le h^{(r)}(k, j+1) \le \cdots \le h^{(r)}(k, l)$

which proves Theorem 1.

Theorem 2. For $i > -1$, $j > -1$ and $r > 1$,

$$h^{(r)}(i, j) \le h^{(r-1)}(i, j) \tag{24}$$

Proof. The proof of Theorem 2 is similar to that of Theorem 1.

Lemma 2. For $i > -1$, $j > -1$and $r > 1$,

$$h^{(r+1)}(i, j+1) \le h^{(r)}(i, j) \tag{25}$$

$$h^{(r+1)}(i+1, j) \le h^{(r)}(i, j) \tag{26}$$

Proof. Lemma 2 can be proven by induction.

Theorem 3. For $r > 1$, $1 \le t \le 8$,

$$h^{(r+2)}\left(i + s_{t,1}, j + s_{t,2}\right) \le h^{(r)}(i, j) \tag{27}$$

$$h^{(r+2)}(i, j) \le h^{(r)}\left(i + s_{t,1}, j + s_{t,2}\right) \tag{28}$$

Proof. From Theorem 2 and Lemma 2, and for $r > 1$, we have

$$h^{(r+2)}\left(i + s_{t,1}, j + s_{t,2}\right) \le h^{(r+2)}(i+1, j+1) \le h^{(r+1)}(i, j+1) \le h^{(r)}(i, j) \tag{29}$$

$$h^{(r+2)}(i, j) \le h^{(r+1)}(i-1, j) \le h^{(r)}(i-1, j-1) \le h^{(r)}\left(i + s_{t,1}, j + s_{t,2}\right) \tag{30}$$

So Theorem 3 is true.

Theorem 4. For $i > -1$, $j > -1$ and $r > 1$, the minimal value of $h^{(r)}(i, j)$ is a real root of equation (31), which is almost equal to 2^{-8n}.

$$[(1 - 2^{-n})\, a + 2^{-n}]^8 = a \tag{31}$$

Proof. Suppose $h^{(r)}(i, j)$ achieves the minimal value a for some $i > -1$, $j > -1$ and $r > 1$, then for any $r' > r$, we have $a \le h^{(r')}(i, j)$. From Theorem 2, $a \le h^{(r')}(i, j) \le h^{(r)}(i, j) = a$. So $h^{(r')}(i, j) = a$. By Theorem 3, we have

$$a = h^{(r+4)}(i, j) \le h^{(r+2)}\left(i + s_{t,1}, j + s_{t,2}\right) \le h^{(r)}(i, j) = a.$$

So $h^{(r+2)}\left(i + s_{t,1}, j + s_{t,2}\right) = a$.

Then we have Equation (31) from (16) and (17). Since $h^{(r)}(i, j)$ is a probability and $a < 1$, a must be one real root of Equation (31) on [0, 1). Only one root satisfies this requirement, whose value is close to 2^{-8n} with a maximal error of 10^{-5}.

Theorem 5. For $i > -1$, $j > -1$, we have $h^{(R)}(i, j) = a$, where a is the real root of Equation **错误!未找到引用源。** on [0, 1) and

$$R = i + j + 7 \tag{32}$$

Proof. Calculation from (14) and (17) shows that for $n > 0$, $h^{(7)}(0,0) = a$. According to Lemma 2, after another i rounds of iterations, we have $h^{(i+7)}(i,0) \leq a$. And another j rounds later, we have $h^{(i+j+7)}(i,j) \leq a$. Since a is the minimal value of $h^{(r)}(i,j)$ for *all* $r > 2$, we have $h^{(i+j+7)}(i,j) = a$, which proves Theorem 5.

$h^{(r)}(i,j)$ is the probability that the cell (i, j) is white in the coloring model, which corresponds to the probability that the pixel keeps unchanged in the same position of the cipher-image after r rounds of evolutions of the PCA. Therefore Theorem 1 to Theorem 5 sketch out how the changes of pixels propagate during evolutions of the PCA model. The probability to change for each pixel tends to increase after each round of evolution. Such probability reaches the maximal value for pixel (i, j) when $h^{(r)}(i,j)$ equals a after several rounds of evolutions; within one more round, the pixels on the Von Neumann neighborhoods of (i, j) also have the maximal probability to change; So do those on Moore neighborhoods within two more rounds. For an $M \times N$ image, all its pixels will change with a probability around $1-2^{-8n}$ in no more than $M+N+7$ rounds.

Since function f fulfills the SAC, once a pixel changes, each of its bit will change with probability of 0.5. If one bit of the plain-image changes, all but about $2^{-8n} \times 100\%$ of the bits in the cipher-image change with a probability of 0.5 in at most $M+N+7$ rounds of encryption. Therefore, the proposed encryption model is considered to satisfy the global SAC.

5.3 Estimation of the minimum number of round to meet global SAC for Square Images

Theorem 5 points out that $M+N+7$ is the theoretical upper bound of the number of rounds required to fulfill the global SAC for an $M \times N$ image. Nevertheless, the encryption model meets the global SAC in much fewer numbers of rounds. In the following sections, we will discuss the approaches to estimate $r_{\mathrm{SAC}}(M, N)$, the minimal number of rounds to satisfy the global SAC for an $M \times N$ image. Note that r_{SAC} determines the computational efficiency of our proposed encryption model, as the encryption time of a pixel can be considered constant in each round.

We start from a special case, the square image. The estimation of $r_{\mathrm{SAC}}(M, N)$ for rectangle image is presented in section 5.4.

Let $r_{M,N}(i,j) = \min\{r | h(i,j,r) = a, r > 1\}$ for an $M \times N$ image. Theorem 1 indicates that $h(i,j,r) = a$ for all $0 \leq i \leq \lfloor M/2 \rfloor, 0 \leq j \leq \lfloor N/2 \rfloor$ when $h\left(\left\lfloor \frac{M}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor, r\right) = a$. So we have

$$r_{\mathrm{SAC}}(M,N) = r_{M,N}\left(\left\lfloor \frac{M}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor\right) \tag{33}$$

To estimate the value of $r_{M,M}\left(\left\lfloor \frac{M}{2} \right\rfloor, \left\lfloor \frac{M}{2} \right\rfloor\right)$ for an $M \times M$ square image, we focus on how the changes of pixels propagate from (0, 0) to $(\left\lfloor \frac{M}{2} \right\rfloor, \left\lfloor \frac{M}{2} \right\rfloor)$ along the diagonal of the image. We denote the $r_{M,M}(i,i)$ as $r_M(i)$ for simplicity. The number of round(s) required to propagate a change from (i, i) to (i+1, i+1) equals

$$\Delta r_i = r_M(i+1) - r_M(i) \tag{34}$$

And the average number of rounds to propagate the changes from (i, i) to (i+1, i+1) is given by

$$\overline{\Delta r} = \frac{1}{l}\sum_{k=i}^{i+l-1} \Delta r_k \tag{35}$$

where l is a positive integer. Therefore, $r_M(i)$ can be approximately calculated by

$$r_M(i) = \left\lfloor i\overline{\Delta r} + r_M(0) \right\rfloor \tag{36}$$

We need to determine the value of $\overline{\Delta r}$ and $r_M(0)$ to calculate $r_{\mathrm{SAC}}(M,M)$. Nevertheless, the values of the two parameters depend on n. In the following discussion, we estimate $r_{\mathrm{SAC}}(M,M)$ in cases of n=1, n=2 and n>2.

1) n=1

In this case, Δr_i is not regular. However, $\overline{\Delta r}$ is stable, which leads to a strong positive linear correlation between $r_{\mathrm{SAC}}(M,M)$ and M (See Fig. 5).

A more detailed observation on coloring model suggests that, usually 7 of 8 consecutive values of Δr_i are equal to 1 and the remaining one equals 2. Therefore, we have

$$\overline{\Delta r} \approx \frac{1}{8}(7\times 1 + 1\times 2) = \frac{9}{8} \tag{37}$$

And $r_M(0)$ is estimated as 3. From **错误!未找到引用源。**, **错误!未找到引用源。** and **错误!未找到引用源。**, we have

$$r_{\mathrm{SAC}}(M,M) \approx \left\lfloor \frac{9}{8}\times\frac{M}{2} + 3 \right\rfloor \tag{38}$$

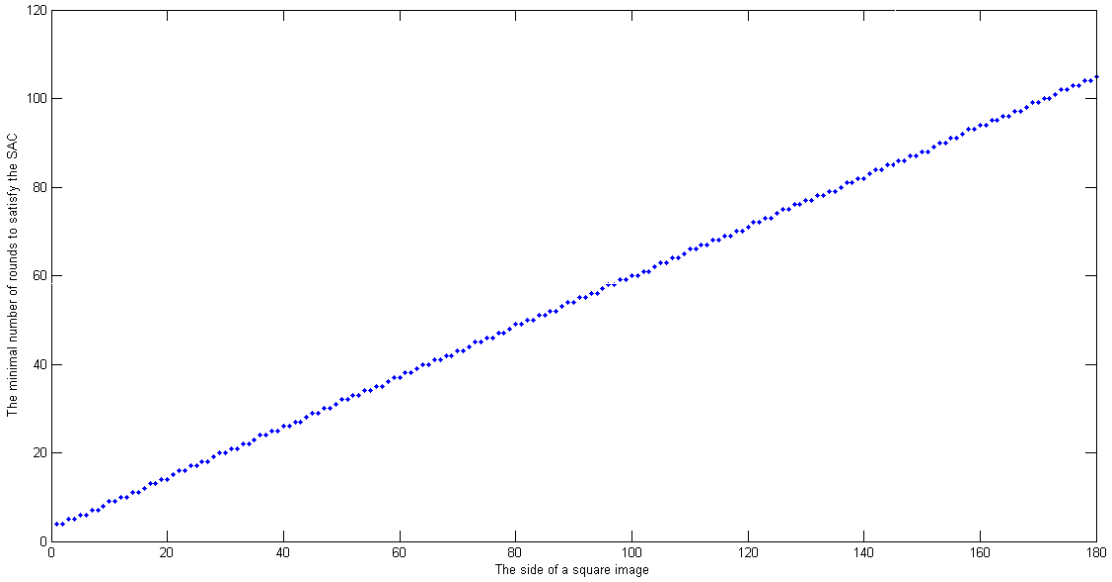


Fig. 5. The correlation between $r_{\mathrm{SAC}}(M,M)$ and M

Simulation on square images with side from 5 to 180 illustrates that (38) can predict the $r_{\mathrm{SAC}}(M,M)$ with an error of no greater than 1 round.

2) n=2

For n=2, we have $\overline{\Delta r} \approx 33/32$ and $r_M(0)$=2. The $r_{\mathrm{SAC}}(M,M)$ is estimated as

$$r_{\mathrm{SAC}}(M,M) \approx \left\lfloor \frac{33}{32}\times\frac{M}{2} + 2 \right\rfloor \tag{39}$$

Simulation shows that the maximal predicted error of (39) is only 1 round.

3) n>2

When $n > 2$, the changes along the diagonal of the square image propagate so fast that Δr_i=1 in overwhelming majority of cases. Thus we have $\overline{\Delta r} \approx 1$ and the $r_{\mathrm{SAC}}(M, M)$ is estimated as

$$r_{\mathrm{SAC}}(M, M) \approx \left\lfloor \frac{M}{2} \right\rfloor + 2 \tag{40}$$

Simulation shows that the predicted error of (40) is at most 1 round.

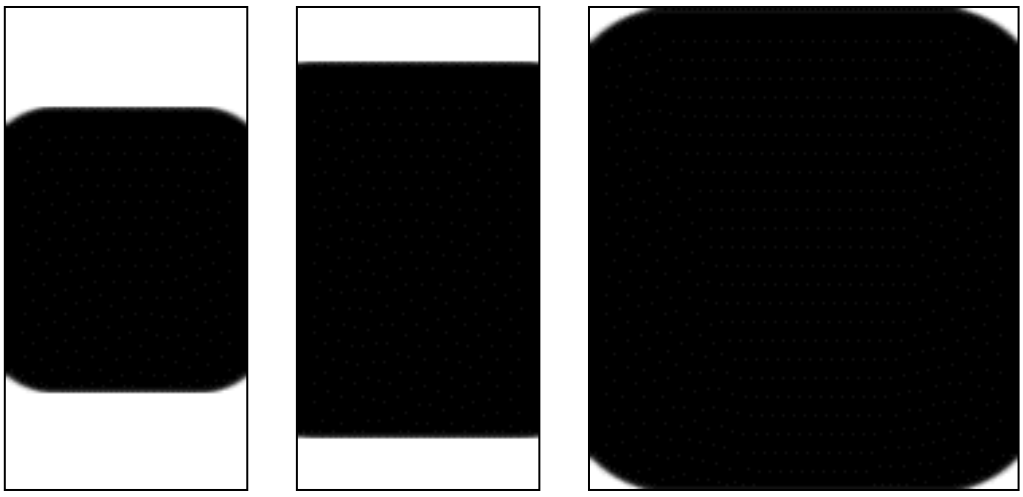
5.4 Estimation of the minimum number of round to meet global SAC for Rectangle Images

In the following discussion, we estimate the value of r_{SAC} for general rectangular images. Since the coloring model is symmetric, we have

$$r_{\mathrm{SAC}}(M, N) = r_{\mathrm{SAC}}(N, M) \tag{41}$$

Let us assume $M \geq N$. We find two approaches to estimate $r_{\mathrm{SAC}}(M, N)$. The first approach (we call Approach I) consumes no extra memory space while its accuracy is still acceptable; the second one (we call Approach II) can estimate $r_{\mathrm{SAC}}(M, N)$ with high accuracy but requires some extra memory space.

In approach I, the propagation of changes from (0, 0) to $(\left\lfloor \frac{M}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor)$ can be divided into three stages (See Fig.6):



(a) (b) (c)

Fig. 6. Illustration of the propagation process, *M*=180. (a) End of the early stage, *N*=90; (b) End of the Intermediate stage, case 1, *N*=90; (c) End of the Intermediate stage, case 2, *N*=160.

(1) Early stage: In this stage, the changes are propagated from (0, 0) to as far as $(\left\lfloor \frac{N}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor)$. The propagation process in this stage is almost the same to that for an *N*×*N* image. And the propagation along the central column (i.e., the 0-th column) is faster than the rightmost column (i.e., the $\left\lfloor \frac{N}{2} \right\rfloor$-th column);

(2) Intermediate stage: In this stage, the propagation of changes along the rightmost column is faster than the central column. Therefore, this stage might end in either of two cases: The propagation along the rightmost column catches up to the central column, or before that happens, the changes

are first propagated to $\left(\left\lfloor\frac{M}{2}\right\rfloor, 0\right)$ along the central column;

(3) Final stage: This stage starts from the end of intermediate stage and ends when the changes spread over the whole image.

Again, we discuss the estimation of $r_{\mathrm{SAC}}(M,N)$ in cases of $n = 1$, $n = 2$ and $n > 2$.

1) $n = 1$

For $n = 1$, the difference between M and N determines in which case the intermediate stage ends. If the difference is large enough, the propagation along the rightmost column will catch up to the central column (See Fig. 6(b)). Then the changes along all columns propagate in a same speed and we have

$$r_{M,N}\left(\left\lfloor\frac{M}{2}\right\rfloor, \left\lfloor\frac{N}{2}\right\rfloor\right) = r_{M,N}\left(\left\lfloor\frac{M}{2}\right\rfloor, 0\right) \tag{42}$$

We observe that the speed of propagation along the central column is almost as regular as 1, i.e., $\overline{\Delta r} \approx 1$. And $r_{M,N}(0,0)$ is around 2. Thus the r_{SAC} of an $M \times N$ image when $M \gg N$ is estimated as:

$$r_{\mathrm{SAC}}(M,N) = r_{M,N}\left(\left\lfloor\frac{M}{2}\right\rfloor, 0\right) = \left\lfloor\frac{M}{2}\overline{\Delta r} + r_{M,N}(0,0)\right\rfloor \approx \left\lfloor\frac{M}{2}\right\rfloor + 2 \tag{43}$$

If N is close to M, the propagation along the central column reaches the topmost row first (See Fig. 6(c)). The speed of the propagation along the rightmost column is relatively stable in the intermediate and final stage. Simulation shows that, the changes spread over 5 rows in 4 rounds usually, i.e., $\overline{\Delta r} \approx 4/5$. The early stage takes about $9N/16+3$ rounds as indicated by (38), and the changes need to spread over about $(M-N)/2$ rows in the intermediate and final stage. So we have

$$r_{\mathrm{SAC}}(M,N) = \left\lfloor\frac{9}{8} \times \frac{N}{2} + 3 + \left(\frac{M}{2} - \frac{N}{2}\right) \times \frac{4}{5}\right\rfloor = \left\lfloor\frac{2}{5}M + \frac{13}{80}N + 3\right\rfloor \tag{44}$$

It is possible that the changes in the central column happen to propagate to the topmost row when the propagation along the rightmost column catches up to the central one. In this particular case, $r_{\mathrm{SAC}}(M,N)$ satisfies both (43)and (44), and we have Equation (45):

$$\left\lfloor\frac{2}{5}M + \frac{13}{80}N + 3\right\rfloor = \lfloor M/2 \rfloor + 2 \tag{45}$$

Solving this equation, we get $N = \lfloor(8M - 80)/13\rfloor$. Therefore, r_{SAC} for an $M \times N$ image is estimated by (46).

$$r_{\mathrm{SAC}}(M,N) \approx \begin{cases} \left\lfloor\frac{M}{2}\right\rfloor + 2 & , \ N \le \frac{8M-80}{13} \\ \left\lfloor\frac{2}{5}M + \frac{13}{80}N + 3\right\rfloor & , \ N > \frac{8M-80}{13} \end{cases} \tag{46}$$

2) $n = 2$

The propagation of changes still comprises of early, intermediate, and final stages for n=2. Nevertheless, the propagation is faster and we have $\overline{\Delta r} \approx 1/2$ in the intermediate stage and final stage. The r_{SAC} of an $M \times N$ image is estimated as follows

$$r_{\mathrm{SAC}}(M,N) \approx \begin{cases} \left\lfloor\frac{M}{2}\right\rfloor & , \ N \le \frac{16M-128}{17} \\ \left\lfloor\frac{1}{4}M + \frac{17}{64}N + 2\right\rfloor & , \ N > \frac{16M-128}{17} \end{cases} \tag{47}$$

3) $n > 2$

When n>2, the propagation is so fast and stable that $\overline{\Delta r} \approx 1$ through all of the three stages, and the

$r_{\mathrm{SAC}}(M,N)$ is estimated as

$$r_{\mathrm{SAC}}(M,N) \approx \left\lfloor \frac{M}{2} \right\rfloor \tag{48}$$

In Approach II, we predict $r_{\mathrm{SAC}}(M,N)$ with reference to the propagation process of the $M \times M$ image. The propagation of changes from (0, 0) to $(\lfloor \frac{M}{2} \rfloor, 0)$ and to $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{N}{2} \rfloor)$ for an $M \times N$ image is considered almost the same to that for $M \times M$ image, since the pixels in these positions have not been affected by the leftmost pixels in most rounds of the encryption. Only in the last several rounds, does the influence of the leftmost pixels pass through the left boundary and accelerate the propagation process. Accordingly, $r_{\mathrm{SAC}}(M,N)$ can be deduced from $r_{M,M}$:

$$r_{\mathrm{SAC}}(M,N) = \begin{cases} \left\lfloor \frac{M}{2} \right\rfloor + 2 & ,0 < N < 2d \\ r_{M,M}\left(\left\lfloor \frac{M}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor - d \right) & ,N \geq 2d \end{cases} \tag{49}$$

where d reflects the extent of the acceleration, and we estimate d as 1 for the coloring model. Fig. 7 shows the practical values of $r_{\mathrm{SAC}}(M,N)$ and the predicted results from (49) with M = 90 and N = 2, 3, …, 90.

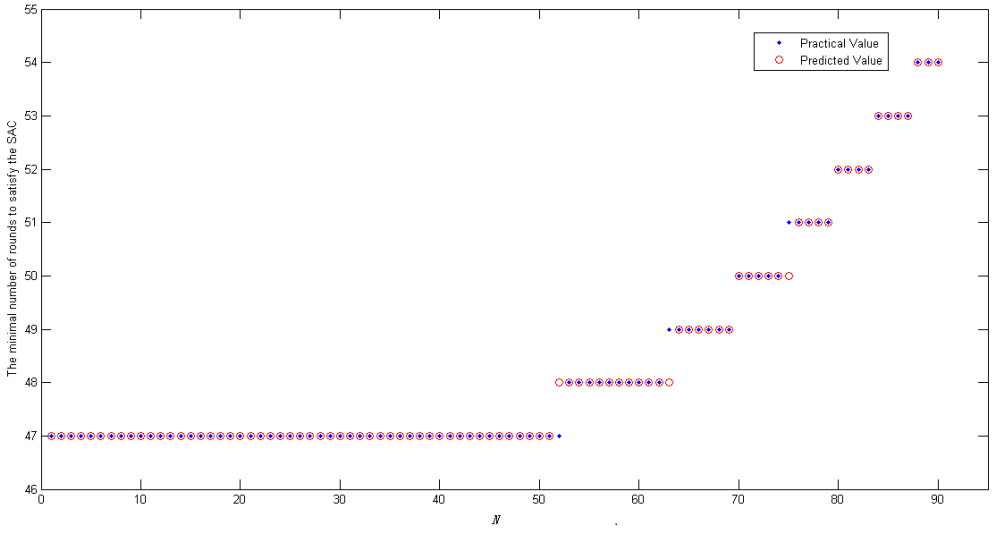


Fig. 7. The practical value of $r_{\mathrm{SAC}}(M,N)$ and that predicted by (49)

Table 3 lists the results in prediction of $r_{\mathrm{SAC}}(M,N)$ by Approach I and II, where n denotes the number of bits of each pixel, M is the number of rows. We predict the value of r_{SAC} by Approach I and II, and compare it with the practical value computed by (33). Table 3 lists the statistics of the distance between the practical and the predicted value of $r_{\mathrm{SAC}}(M,N)$, where N denotes the number of columns varying from 1 to M. It shows that the Approach II has a better performance in the estimation of r_{SAC}. The Approach I is also effective with a maximal error of 2 rounds. Besides, unlike its counterpart, Approach I requires no reference to $r_{M,M}$, which needs to be computed and stored beforehand in Approach II.

Table 3. Predication of r_{SAC} for rectangular images

n	M	Approach	Accuracy	Max Distance	Average Distance
1	90	I	84.44%	1	0.1556
1	90	II	96.67%	1	0.0333
1	180	I	68.89%	2	0.3667
1	180	II	94.44%	1	0.0556
2	180	I	91.67%	2	0.0944
2	180	II	93.33%	1	0.0667
3	180	I	95.00%	2	0.0722
3	180	II	96.11%	1	0.0389

6. PCA-Based Image Encryption Algorithm and Its Implementation

基于PCA的图像加密算法及其实现

We propose two image encryptions algorithms based on PCA. The first one uses von Neumann neighborhood, which fails to meet global SAC. The second one is designed in accordance with the model proposed in section 4. The round key expansion algorithm, software and hardware implementation for the second algorithm are also presented. Its experimental results are given in next section.

6.1 Image Encryption Algorithm Using Von Neumann Neighborhood

基于冯诺依曼邻域的图像加密算法

The Von Neumann neighborhood is widely adopted in the digital image processing and computer graphics techniques. Here, we test the sensitivity of the encryption models using Von Neumann Neighborhood. The neighborhoods from s_1 to s_4 are (-1, 0), (0, 1), (1, 0) and (0, -1) respectively. The function f is implemented by a pseudo-random function satisfying SAC.

We choose 4 MSBs (Most Significant Bit-planes) of 128×128 'Lena' as the first plain-image I_1, and produce the second plain-image I_2 by changing one bit of a pixel in I_1. Then C_1 and C_2, the cipher-images corresponding to I_1 and I_2, are generated after 128 rounds of evolutions of the PCA. We then compare the difference between C_1 and C_2. Table 4 lists the change rates of bits and pixels (i.e., the percentage of bits and pixels different in C_1 and C_2) and their expected values according to the global SAC. Table 4 indicates that the encryption model using Von Neumann Neighborhood is substandard with regard to the SAC.

Table 4. Bit and pixel change rates and their expected values

	Practical value	Expected value (m = 4)
Bit change rate	24.93%	50%
Pixel change rate	46.97%	93.75%

Why cannot the model using Von Neumann Neighborhood meet the global SAC? Since the change of a cell only affects its orthogonally surrounding cells, the even cells (We call the cell (i, j) even cell if $i + j$ is an even number, otherwise we call it odd cell) only affects the odd cells and vice visa. Suppose an even cell changes at first, then only odd cells change after the model evolves for one round, and later the even cells and the odd cells change alternatively. Fig. 8 depicts the difference between the pixels in the upper left corner of C_1 and C_2, where only the even cells change (those colored in black). Since the percentage of the pixels that change will never exceed 50% (while the expected percentage is $(1-2^{-4})$ ×100% = 93.75%), the model using Von Neumann Neighborhood cannot meet the global SAC.

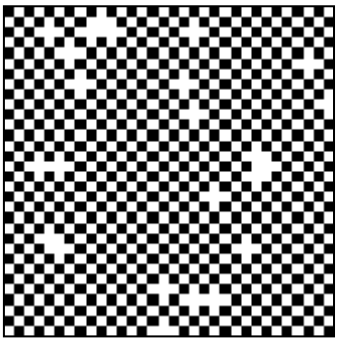


Fig. 8. The difference of some pixels between C_1 and C_2

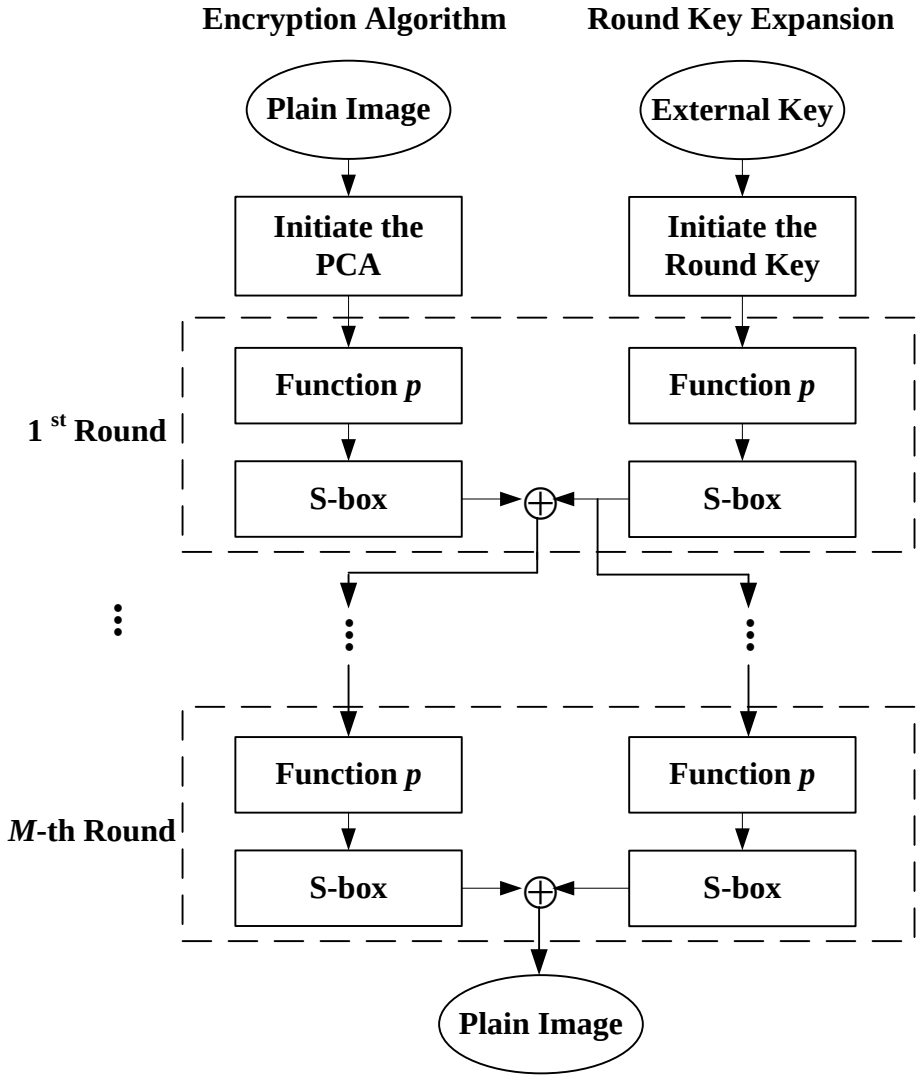


Fig. 9. The encryption algorithm and the round key expansion

基于加密模型的图像加密算法

6.2 Image Encryption Algorithm Based on the Proposed Encryption Model

Here, we propose an image encryption algorithm based on the encryption model presented in Section 4. We set $n = 1$ so that the algorithm is suitable for grayscale image. We also describe the way to generate the round key from an external key and to use the round key in each round. Then we

discuss the implementation issues of this algorithm on both software and hardware. The experiment results will be presented in next part.

The proposed PCA model involves two functions, p and f, corresponding to permutation and substitution operations, respectively. Since function p is fixed, we focus on the design of function f. We demand function f fulfill the following requirements:

(1) It is a reversible function;

(2) Each Boolean function corresponding to f is near-perfect nonlinear;

(3) It can accommodate the round key.

Besides, the function f should be kept as simple as possible for the sake of high efficiency in encryption. Therefore, function f consists of only two operations, a bijective S-box substitution followed by an XOR operation with the round key (See Fig. 9). Both operations are reversible, and the round key is introduced by the XOR operation. Since the XOR operation is linear, it has no influence on the nonlinearity of S-box. Nevertheless, S-box must be carefully selected to ensure that it is highly nonlinear.

6.3 Round Key Expansion 轮密钥扩展

The image encryption algorithm accepts 128-bit external key, while the encryption of an $M \times N$ grayscale image needs $M \times N$ bytes of data as the round key. Moreover, the round key should be different for each round. Here, we describe the way how the fixed-length external key expands to the same size of the plain-image and how it evolves to a new round key.

Initiation of the round key: In the initial stage, the 128-bit external key is first arrayed to a 4×4 grayscale image K, then the first round key is created by $M/4 \times N/4$ tiling of the copies of K.

Evolution of the round key: The evolution of the round key is similar to that of the cipher-image, which consists of function p and an S-box. Then the round key is then XORed with the cipher-image.

Fig. 9 also shows the initiation and evolution of the round key, and how it participates the encryption of the image. The evolutions of the cipher-image and of the round key are kept symmetric so that they can be executed in parallel.

6.4 Software and Hardware Implementation 软硬件实现

The encryption algorithm consists of three operations: function p, S-box and XOR. The S-box and XOR is easy and direct to implement by both software and hardware. However, the implementation of function p is a little tougher. If we treat the grayscale image as 8 bit-planes, then the operations are uniform on each bit of a particular bit-plane. And the operation on the bits of the t-th bit-plane can be represented as

$$I'_t(i, j) = I_t((i + s_{t,1}) \bmod M, (j + s_{t,2}) \bmod N) \tag{50}$$

where $I_t(i, j)$ and $I'_t(i, j)$ denote the value of the bit on i-th row and j-th column of the t-th bit-plane before and after the operation of function p, respectively. For example, the operations on the bits of the first bit-plane are essentially left circular shift ($s_1 = (0, 1)$).

The software implementation of function p demands two identical memory spaces to support the parallel computing effectively. Each space has a same size as the cipher-image. The cipher-images before and after the operation of function p are stored in two spaces alternatively. The hardware version of function p is easier than the software one, where the operation on each bit-plane is implemented by arrays of D flip-flops. Fig. 10 shows the hardware implementation of function p on the first bit-plane of the image.

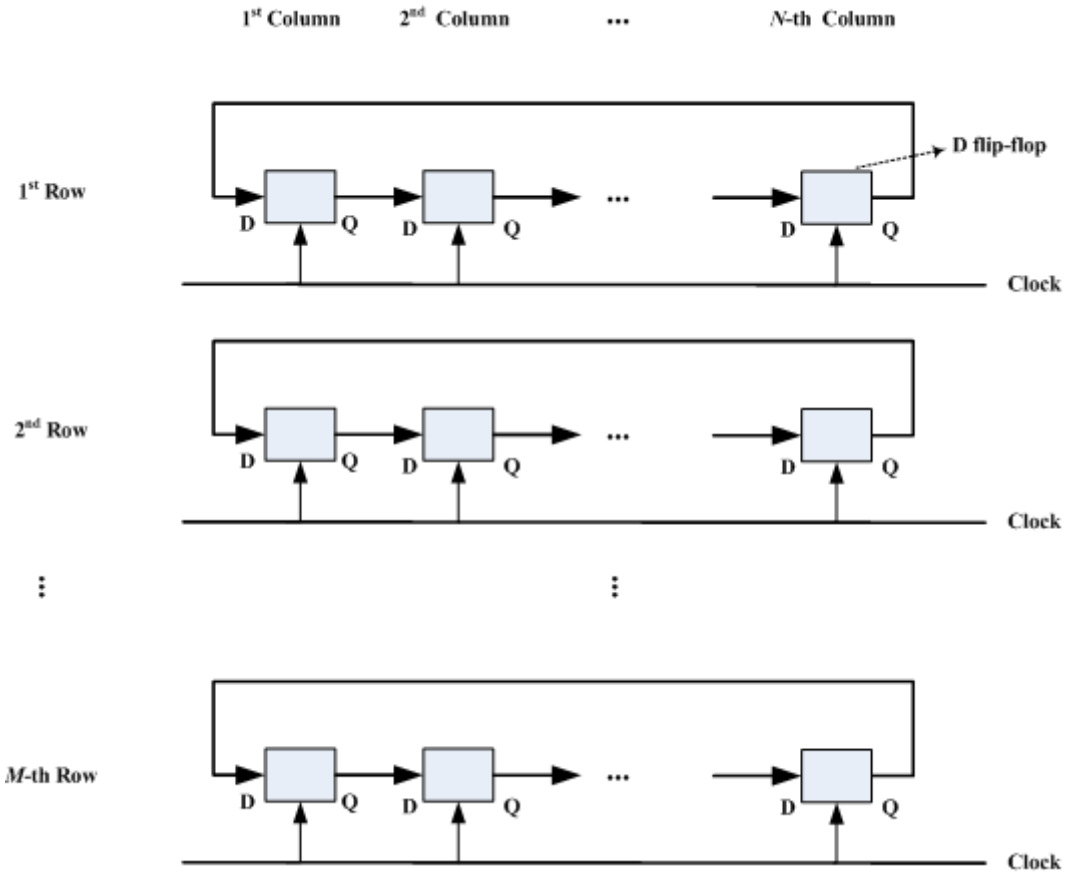


Fig. 10 Hardware implementation of function p on the first bit-plane of the image

7. Experimental Results

Numerical simulation is of essence to verify the effectiveness of a theoretic model [34-35]. We simulated the image encryption algorithm on personal computer to check its properties of randomness and sensitivity. Besides, we verified whether the approach II proposed in Section 5 could predict r_{SAC} correctly for a practical encryption algorithm based on PCA.

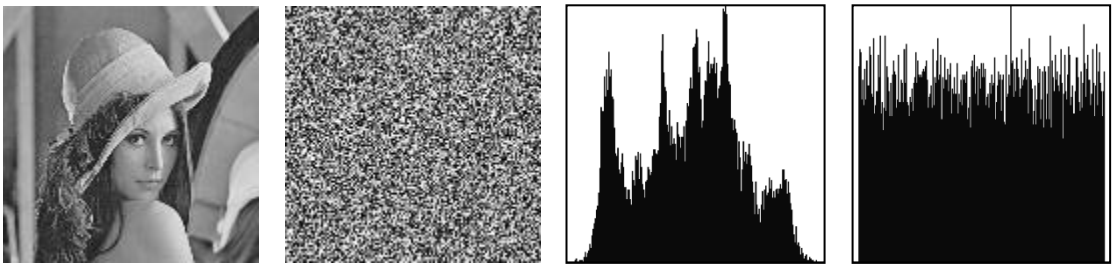
In the experiment, we focus on the case of n = 1. S-box is chosen as the same one used in AES standard [36]. It can be represented by 8 Boolean functions f_1, …, f_8. $f_j : \{0,1\}^8 \to \{0,1\}, j = 1,...,8$. Table 5 lists the statistics of $r(a)$ for these Boolean functions. It shows that all Boolean functions are near-perfect nonlinear.

Table 5. Statistics of r(a) of 8 Boolean functions for AES S-box

Maximum	Minimum	Mean
0.5625	0.4375	0.5020

The images we chose as the plain-images include 14 frequently used images from USC-SIPI image database [37], and some images generated by computer program. We only present the experiment results on 128×128 Lena image in most cases when these results are representative of those on other images. The external key is set to 128 bits of zeros, and the number of encryption round equals the larger one of M and N. Fig. 11 shows the plain-image and

cipher-image.



(a) (b) (c) (d)

Fig. 11. Plain-image, cipher-image, and their histograms. (a) Plain-cipher (b) cipher-image (c) histogram of the plain-image (d) histogram of the cipher-image

7.1 Evaluation of Randomness Property

随机性评价

Fig. 11 also shows the histogram of the plain-image and the cipher-image. The histogram of cipher-image is flat in contrast to that of the plain-image.

The percentage of the bits equal to 1 in the cipher-image of Lena for each round is depicted by Fig. 12. The percentage is stable around 50% after 5 rounds of encryption.

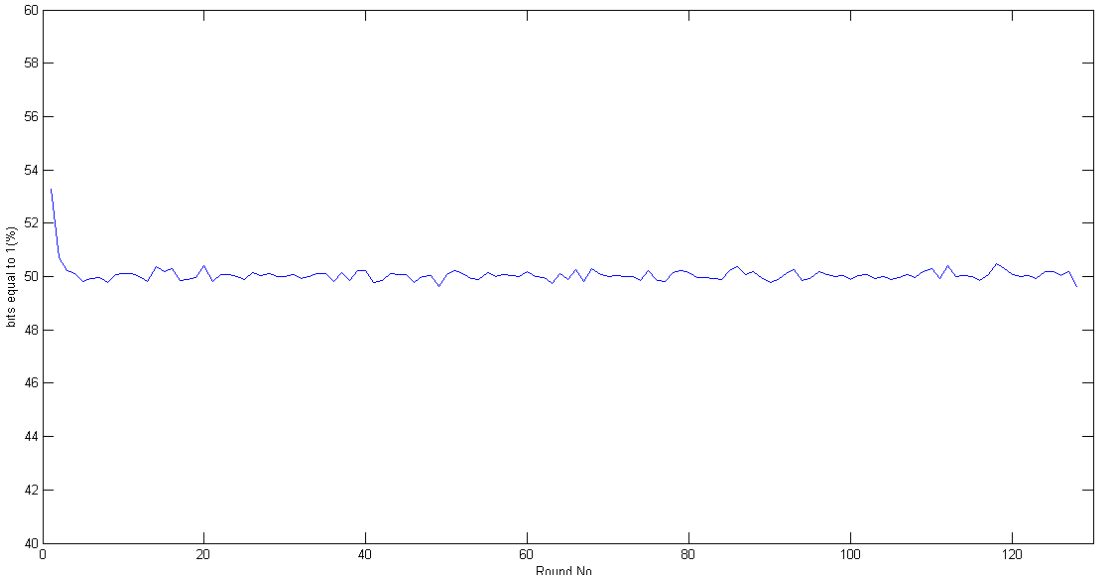


Fig. 12. The percentage of the bits equal to 1 in 128 rounds of encryption

Table 6 lists the same percentages for cipher-image of 14 images selected from USC-SIPI image database, all of which are close to 50%.

Table 6. The percentages of bits equal to 1 for cipher-images of 14 images (%)

Airplane	**Baboon**	**Couple**	**Girl01**	**Girl03**	**Girl04**	**House05**
49.99	49.96	49.79	49.89	50.02	50.26	50.18
House	**JellyBeans07**	**JellyBeans08**	**Lena**	**Peppers**	**Lake**	**Splash**
49.92	49.62	49.98	50.16	50.00	49.88	50.10

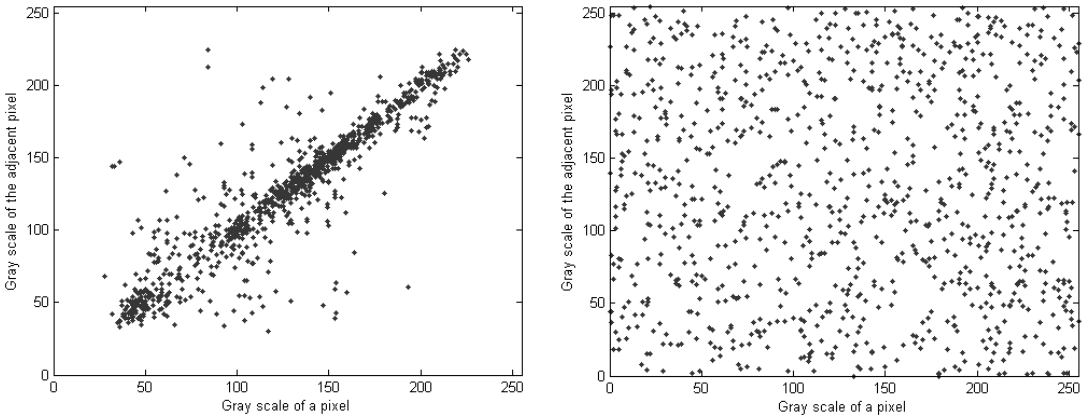
We also created 1000 random images by computer program. The percentages of bits equal to 1 in these plain-images are from 1% to 99%. Table 7 lists the statistics of the percentages of bits equal to 1 in their corresponding cipher-images. The result illustrates that the distribution of 0 and 1 in these

cipher-images is uniform.

Table 7. Statistics of the percentages of bits equal to 1 for 1000 images

Maximum	Minimum	Mean	Standard deviation
50.49%	49.61%	50.01%	0.0013

Another indicator of the randomness of cipher-image is the correlation between adjacent pixels, which is usually close to 1 for common plain-images. On the contrary, such correlation is around 0 for cipher-image with excellent property of randomness. Fig. 13 shows the grayscales of 1000 pairs of adjacent pixels in horizontal directions for both Lena and its cipher-image (The x and y coordinates of each point are the grayscales of two horizontally adjacent pixels). Table 8 lists the correlation coefficients of adjacent pixels of the plain-image and cipher-image in horizontal, vertical and diagonal directions. They show that no evident correlation is found between adjacent pixels in the cipher-image.



(a) (b)

Fig. 13. Correlation of two horizontally adjacent pixels (a) plain-image (b) cipher-image

Table 8. Correlation coefficient of two adjacent pixels in plain-image and cipher-image.

Direction	Plain-image	Cipher-image
Horizontal	0. 9403	-0. 0136
Vertical	0. 9678	-0. 0089
Diagonal	0. 9218	-0. 0100

7.2 Evaluation of Sensitivity Property 灵敏度特性评价

In order to test the sensitivity of the image encryption algorithm, the plain-image or the external key is changed with only one bit and then the corresponding cipher-images are compared. As indicated by the global SAC criterion, about 50% of the cipher-image bits are expected to change. Fig. 14 shows the change rate for each round when the plain-image and the external key are changed with one bit. It seems that the cipher-image is more sensitive to the external key than to the plain-image, since one bit of change in the external key results in many bits of change in the round key. The test results of sensitivity for 14 images from USC-SIPI image database and for 1000 random images are listed in Table 9 and

Table 10. All the change rates are close to 0.5.

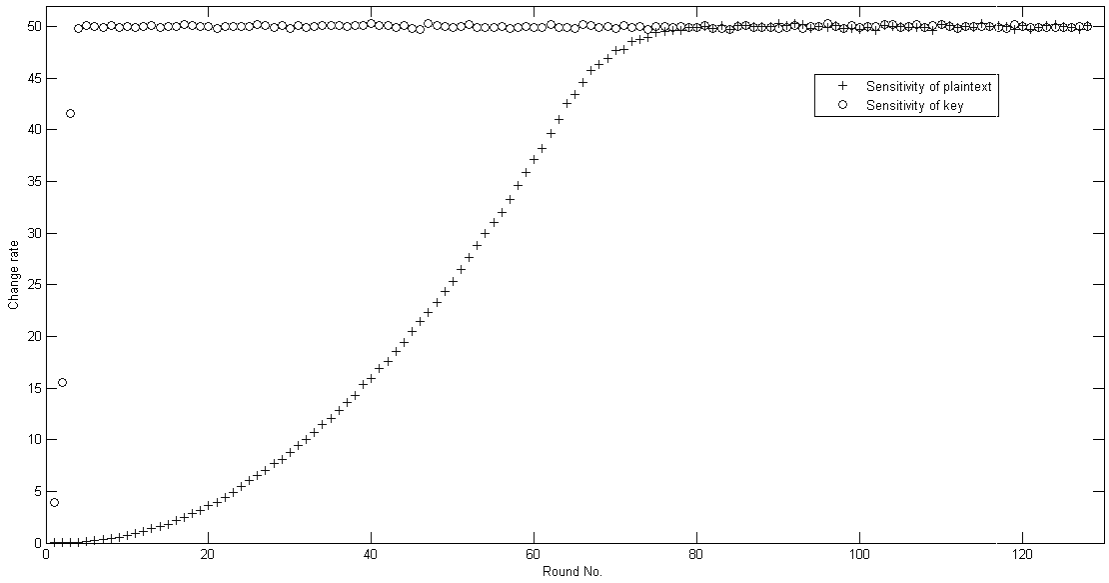


Fig. 14. The sensitivity of the encryption algorithm to plain-image and key

Table 9. The change rates of cipher-bits for 14 images

Airplane	**Baboon**	**Couple**	**Girl01**	**Girl03**	**Girl04**	**House05**
49.83	49.63	50.00	50.09	49.87	49.88	50.04
House	**JellyBeans07**	**JellyBeans08**	**Lena**	**Peppers**	**Lake**	**Splash**
49.93	50.01	49.96	50.07	50.08	50.01	49.80

Table 10. The statistics of the change rates for 1000 images

maximum	**minimum**	**mean**	**standard deviation**
50.47%	49.54%	50.01%	0.0014

Another indicator of the sensitivity of image encryption algorithm is the average distance of pixels of cipher-images corresponding to two similar plain-images. Given an image *P*, we create another image *P*' by flipping one bit of *P*. The average distance between their corresponding ciphertexts *C* and *C*' is defined as

$$D = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|C_{i,j} - C'_{i,j}\right| \tag{51}$$

where $C_{i,j}$, $C'_{i,j}$ is the grayscale of the pixel on the *i*-th row and *j*-th column of image *C* and *C*', respectively. For grayscale images, the distance is expected to around

$$\mathrm{Exp}(D) = \frac{1}{256^2}\sum_{x=0}^{255}\sum_{y=0}^{255}|x-y| = \frac{65535}{768} \approx 85.3320 \tag{52}$$

Table 11 lists the average distances for 14 images, which shows that all of the practical distances are close to the theoretical value.

Table 11. The practical distance for 14 images

Airplane	Baboon	Couple	Girl01	Girl03	Girl04	House05
85.06	85.20	85.94	85.67	85.47	84.73	85.57
House	**Jelly Beans07**	**Jelly Beans08**	**Lena**	**Peppers**	**Lake**	**Splash**
84.93	85.71	85.41	85.19	85.10	84.96	85.63

7.3 Estimation of the minimum number of round for global SAC

We tested the accuracy of approach II when it is applied to the proposed encryption algorithm. We use (49) to predict the value of $r_{\mathrm{SAC}(M,N)}$ from $r_{M,M}(i,j)$). Unlike the coloring model, the practical value of $r_{M,N}(i,j)$ for a practical algorithm cannot be obtained from the evolutions of the model. But we can determine it from a statistical perspective.

Given 100 $M \times N$ images $I_1, \ldots, I_{100}$, we create another 100 images $J_1, \ldots, J_{100}$ by changing one bit of each of $I_1, \ldots, I_{100}$. If, of all the encryptions of 100 pairs of images, there are more than 95 pairs of cipher-images of which the pixel (i, j) changes after at least r rounds of encryption, r is consider as the practical value of $r_{M,N}(i,j)$. Put it in a formal way,

$$r_{M,N}(i,j) = \min\left\{r \middle| \#\left\{k \middle| C_k^{(r)}(i,j) \neq D_k^{(r)}(i,j)\right\} > 95\right\} \tag{53}$$

where $C_k^{(r)}(i,j)$ and $D_k^{(r)}(i,j)$ denote the grayscale of the pixel in the position of (i, j) after r rounds of encryption of the I_k and J_k, respectively. By this means, we can obtain the practical value of $r_{\mathrm{SAC}(M,N)}$ (which equals $r_{M,N}(\lfloor M/2 \rfloor, \lfloor N/2 \rfloor)$) and $r_{M,M}(i,j)$.

Now we can use **错误!未找到引用源。** to predict the value of $r_{\mathrm{SAC}}(M, N)$. (We notice that the speed of propagation of the proposed encryption algorithm is a little faster than that of the ideal coloring model in last several rounds, so we set $d = 4$ here.) Table 12 lists the practical and the predicted values of $r_{\mathrm{SAC}(M,N)}$ with $M = 128$, which shows that $r_{\mathrm{SAC}}(M,N)$ can be predicted with high accuracy.

Table 12. The practical and the predicted value of $r_{\mathrm{SAC}(128,N)}$

N	30	40	50	60	70	80	90	100	110	120
Practical value	66	66	68	68	69	71	73	75	77	80
Predicted value	66	67	67	68	69	70	73	75	77	80

8. Conclusion

Image encryption techniques are expected to achieve better security and efficiency than naïve approaches by taking advantage of the characteristics of digital image. Most space-domain image encryption algorithms can hardly support parallel computing. In this paper, we proposed an image encryption model based on PCA, which is suitable for image encryption and VLSI implementation. Besides, the model is reversible, efficient and can support parallel computing effectively.

The analysis of the security of the encryption model is focused on its ability to satisfy the global SAC. We use the coloring model, a probability CA, to study the global SAC for the encryption model. We show that the theoretical upper bound of the number of rounds required to fulfill the global SAC for an $M \times N$ image is $M+N+7$. Then we proposed several approaches to estimate r_{SAC},

图像加密技术有望实现比Naï有更好的安全性和效率
利用数字图像特征的方法。大多数空间域图像
加密算法难以支持并行计算。在本文中，我们提出了一个图像。
基于PCA的加密模型，适用于图像加密和VLSI实现。
此外，该模型是可逆的，高效的，可以有效地支持并行计算。

the minimal number of rounds, to meet the global SAC for square and rectangular images. As for the rectangular images, the first approach consumes less memory space than the second one; the second approach, however, can predict r_{SAC} with a higher accuracy. Both approaches can estimate the r_{SAC} with a maximal error of one round.

Based on the PCA model, we present an image encryption algorithm and a round key expansion algorithm. The evolutions of the cipher-image and of the round key are kept symmetric so that the two processes can share common structures and be executed in parallel. Experiment results show that the algorithm exhibits satisfactory randomness and sensitivity.

Acknowledgements

The work described in this paper was supported by grants from the National Natural Science Foundation of China (No. 61472464), the Natural Science Foundation of CQ CSTC (No. cstc2016jcyjA0276, cstc2015jcyjA0554, cstc2015jcyjA40025), the fundamental Research Funds for the Central Universities(106112016CDJXY180006) and the National Social Science Foundation of China (No. 14CTQ026)

Reference

[1] C. Yan, H. Xie, D. Yang, J. Yin, Y. Zhang, & Q. Dai, "Supervised hash coding with deep neural network for environment perception of intelligent vehicles." IEEE Transactions on Intelligent Transportation Systems (2017).

[2] L. Zhang, Y. Zhang, J. Tang,X. Gu, J. Li, & Q. Tian, "Topology preserving hashing for similarity search."Proceedings of the 21st ACM international conference on Multimedia. ACM (2013):123-132.

[3] C. Yan, H. Xie, S. Liu, J. Yin, Y. Zhang, & Q. Dai,"Effective Uyghur language text detection in complex background images for traffic prompt identification." IEEE Transactions on Intelligent Transportation Systems (2017).

[4] X. Zhang, H. Zhang, Y. Zhang, *et al.*, "Deep fusion of multiple semantic cues for complex event recognition." IEEE Transactions on Image Processing 25.3 (2016): 1033-1046.

[5] H. Yao, S. Zhang, Y. Zhang, J. Li, & Q. Tian, "Coarse-to-fine description for fine-grained visual categorization." IEEE Transactions on Image Processing 25.10 (2016): 4858-4872.

[6] B. Furht, and D. Kirovski, eds. Multimedia security handbook. CRC press, 2004.

[7] X. Y. Ruan, and R. S. Katti, "A new source coding scheme with small expected length and its application to simple data encryption," IEEE Transactions on Computers 55.10 (2006): 1300-1305.

[8] W. Liu, W. J. Zeng, L. N. Dong *et al.*, "Efficient compression of encrypted grayscale images."IEEE Transactions on Image Processing 19.4 (2010): 1097-1102.

[9] H. Kim, J. T. Wen, and J. D. Villasenor,"Secure arithmetic coding." IEEE Transactions on Signal processing 55.5 (2007): 2263-2272.

[10] M. Podesser, H. Schmidt, and A. Uhl,"Selective bitplane encryption for secure transmission of image data in mobile environments." CD-ROM Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG 2002). (2002):1-6.

[11] J. Fridrich,"Symmetric ciphers based on two-dimensional chaotic maps."International Journal of Bifurcation and chaos 8.06 (1998): 1259-1284.

[12] G. R. Chen, Y. B. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps." Chaos, Solitons & Fractals 21.3 (2004): 749-761.

[13] S. G. Lian, J. S. Sun, and Z. Q. Wang, "A block cipher based on a suitable use of the chaotic standard map." Chaos, Solitons & Fractals 26.1 (2005): 117-129.

[14] S. S. Maniccam, and N. G. Bourbakis, "Image and video encryption using SCAN patterns." Pattern Recognition 37.4 (2004): 725-737.

[15] A. J. Menezes, P. C. Oorschot, and S. A. Vanstone, Handbook of applied cryptography. CRC press, 1996.

[16] C. Yan,Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai, & F. Wu, "A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors." IEEE Signal Processing Letters 21.5 (2014): 573-576.

[17] Q. Zhou, K. Wong, X. F. Liao *et al.*, "Efficient parallel framework for HEVC motion estimation on many-core processors." IEEE Transactions on Circuits and Systems for Video Technology 24.12 (2014): 2077-2089.

[18] Q. Zhou, K. Wong, X. F. Liao *et al.*, "Parallel image encryption algorithm based on discretized chaotic map." Chaos, Solitons & Fractals 38.4 (2008): 1081-1092.

[19] S. Wolfram, "Cryptography with cellular automata." Conference on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, (1985):429-432.

[20] S. Nandi, B. K. Kar, and P. P. Chaudhuri, "Theory and applications of cellular automata in cryptography." IEEE Transactions on computers 43.12 (1994): 1346-1357.

[21] P. Guan, "Cellular automaton public-key cryptosystem." Complex Systems 1.1 (1987): 51-56.

[22] M. Tomassini, M. Sipper, and M. Perrenoud, "On the generation of high-quality random numbers by two-dimensional cellular automata." IEEE Transactions on computers 49.10 (2000): 1146-1151.

[23] S. Das, and B. K. Sikdar, "A scalable test structure for multicore chip." IEEE transactions on computer-aided design of integrated circuits and systems 29.1 (2010): 127-137.

[24] X. Zhao, Q. Li, M. Xu *et al.*, "A symmetric cryptography based on extended cellular automata." Systems, Man and Cybernetics, 2005 IEEE International Conference on. Vol. 1. IEEE, (2005):499-503.

[25] R. Chen, and L. J., "Image security system using recursive cellular automata substitution." Pattern Recognition 40.5 (2007): 1621-1631.

[26] J. Jun, "Image encryption method based on elementary cellular automata."Southeastcon, 2009. SOUTHEASTCON'09. IEEE. IEEE, (2009):345-349.

[27] G. Alvarez, A. Hernandez, L. Hernandez *et al.*, "A new graphic cryptosystem based on one-dimensional memory cellular automata." Security Technology, 2005. CCST'05. 39th Annual 2005 International Carnahan Conference on. IEEE, (2005):200-203.

[28] F. Maleki, A. Mohades, S. M. Hashemi *et al.*, "An image encryption system by cellular automata with memory." Availability, Reliability and Security, 2008. ARES 08. Third International Conference on. IEEE, (2008):1266-1271.

[29] L. Yu, X. Li, and X. Xia, "Image encryption algorithm based on self-adaptive symmetrical-coupled toggle cellular automata." Image and Signal Processing, 2008. CISP'08. Congress on. Vol. 3. IEEE, (2008):32-36.

[30] K. Morita, "Reversible computing and cellular automata—A survey."Theoretical Computer Science 395.1 (2008): 101-131.

[31] C. E. Shannon, "Communication theory of secrecy systems." Bell Labs Technical Journal 28.4 (1949): 656-715.

[32] A. F. Webster, and S. E. Tavares, "On the design of S-boxes."Conference on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, (1985):523-534.

[33] W. Meier, and O. Staffelbach, “Nonlinearity Criteria for Cryptographic Functions,” Lecture Notes in Computer Science, vol. 434,(1990): 549-562,.

[34] O. A. Arqub, & Z. Abo-Hammour, "Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm." Information sciences 279 (2014): 396-415.

[35] O. A. Arqub, M. Al-Smadi, S. Momani, & T. Hayat, "Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems." Soft Computing (2016): 1-16.

[36] Fips, N, "Announcing the ADVANCED ENCRYPTION STANDARD (AES)." National Institute of Standards and Technology (NIST) (2001), http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf29.8(2001):2200-2203.

[37] A.Weber, "The USC-SIPI image database. Signal and Image Processing Institute of the University of Southern California." URL: http://sipi. usc. edu/services/database (1997).