Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 1621–1631
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Image security system using recursive cellular automata substitution

Rong-Jian Chen*, Jui-Lin Lai

Department of Electronics Engineering, National United University, No. 1, Lien Da, Kung-Ching Li, Miaoli 360, Taiwan, ROC

Received 11 December 2005; received in revised form 6 November 2006; accepted 7 November 2006

Abstract

This paper presents a novel image security system based on the replacement of the pixel values using recursive cellular automata (CA) substitution. This proposed image encryption method exhibits the properties of confusion and diffusion because of the characteristics of CA substitution are flexible. The salient features of the proposed image encryption method are its losslessness, symmetric private key encryption, very large number of secret keys, and key-dependent pixel value replacement. Simulation results obtained using some color and gray-level images clearly demonstrate the strong performance of the proposed image security system.

Keywords: Image processing; Cellular automata; Image security; Encryption; Decryption

1. Introduction

As the field of multimedia applications grows, security is increasingly important issue in the communication and storage of images. Encryption is an effective means for ensuring reliable security. Image encryption has been applied to Internet-based communications, multimedia systems, medical imaging, telemedicine and military communication. Numerous image encryption methods are available. They include SCAN-based methods [1–4], chaos-based methods [5–7], tree structure-based methods [8–10] and other systematic methods [11–14]. Each has its strengths and limitations in terms of security, speed and resulting stream size metrics. A new encryption method is proposed to overcome these problems.

The proposed image security system is based on the replacement of the pixel values. Such replacement is made using a recursive cellular automata (CA) substitution with a CA key-stream sequence that is generated using the CA evolution rules. CA has the following advantages: (1) CA has been applied successfully to several physical systems, processes and scientific problems that involve local interactions, as in image processing [15,16], data encryption [17,18] and byte error correcting codes [19]; it has also been used in pseudorandom number generators for built-in VLSI self-tests [20]. (2) The number of CA evolution rules is very large. Hence, many techniques are available for producing a sequence of CA data encrypting and decrypting images. (3) Recursive CA substitution only requires integer arithmetic and/or logic operations, simplifying the computation.

The proposed image security system belongs to the general framework called iterated product cipher [4,21,22], which is based on repeated and intertwined applications of substitution. This general framework has been extensively studied and developed in terms of cryptographic strengths and attacks [4] and forms the basis of many modern encryption methods, including Data Encryption Standard [21,22], Advanced Encryption Standard [23] and chaos-based methods [5–7]. The proposed image encryption method differs markedly from that used elsewhere [17]. In this work, hybrid 2-D *von Neumann* CA was used to generate a high-quality random sequence as the key-stream, with recursive CA substitution in the encryption and decryption schemes, such that the proposed image encryption/decryption system was secure (as shown in Section 5). The cipher systems in the cited study [17] are affine and based on 1-D CA, and the encryption and the decryption schemes in Ref. [17] are non-recursive. Another study [24] showed that affine cipher systems are insecure. The general idea of the recursive substitution has been reported elsewhere [2,4], which are based on

* Corresponding author. Tel.: +886 37 381509; fax: +886 37 362809.
E-mail addresses: rjchen@nuu.edu.tw (R.-J. Chen), jllai@nuu.edu.tw (J.-L. Lai).

0031-3203/$30.00
doi:10.1016/j.patcog.2006.11.011

SCAN methodologies [25]. The proposed recursive CA substitution is a modified and improved version of that presented in cited studies [2,4] because the proposed one used an algorithm of CA-based recursive substitution to enhance the performances of system. The image security system proposed herein has additional features, such as key-dependent pixel value replacement, very large key space, keys of variable length and encryption of larger blocks. Moreover, it is a symmetric private key security system, meaning that the same key is required for encryption and decryption. Therefore, both sender and receiver must know the key.

The rest of this paper is organized as follows: Section 2 provides the background of CA. Section 3 then presents the proposed image encryption/decryption method. Next, Section 4 discusses simulation results. Section 5 gives the possible secret keys, cryptanalysis and comparisons of the performances with other published algorithms. Conclusions are finally drawn in Section 6.

2. CA

CA are dynamic systems in which space and time are discrete. The cells, as arranged in a regular lattice structure, have a finite number of states. These states are updated synchronously according to a specified local rule of neighborhood interaction. The neighborhood of a cell refers to the cell and some or all of its immediately adjacent cells. In 2-D CA space, the specified cell P with its immediate North, South, West and East cells form the *von Neumann* neighborhood; the so-called *Moore* neighborhood incorporates the *von Neumann* neighborhood and the diagonal cells. The states are updated synchronously in discrete time steps for all cells, by applying a specified rule of neighborhood. For a k-state CA, each cell can take any of the integer values between 0 and $(k-1)$.

In a 2-D square space of $N \times N$ cells, the 2-D generalized CA $A(i, j, k, l)$ is given by $\mathbf{a} = a(i, j, t)$, $0 \leqslant i, j \leqslant N-1$, $t \geqslant 0$. For a 2-D *von Neumann* two-state CA, each cell has $2^{2^5} = 2^{32}$ possible CA evolutions, which can be expressed as

$$a(i, j, t+1) = f(a(i-1, j, t), a(i+1, j, t), a(i, t), a(i, j-1, t), a(i, j+1, t)). \quad (1)$$

Here, $f(\cdot)$ is a Boolean function that defines the rule. Wolfram [26] assigned an integer R, the *rule number*, to the Boolean function $f(\cdot)$. In each time step, each cell calculates a new state using Eq. (1). Different cells apply different rules, making the CA *hybrid*. Given that a two-state $N \times N$-cell *von Neumann* 2-D hybrid CA runs over T time steps, it has $2^{2^5 \times N^2} = 2^{32 \times N^2}$ rules, $2^{N \times N}$ initial configurations, 2^{4N} boundary conditions, and results in $2^{32 \times N^2 + N^2 + 4N}$ possible CA evolutions to generate a $T \times N$ N-bit generalized CA data sequence.

Eq. (1) states that the evolution of the (i, j)th cell of a 2-D *von Neumann* two-state CA can be represented as a Boolean function of the present states of the $(i-1, j)$th, $(i, j-1)$th, (i, j)th, $(i, j+1)$th, and $(i+1, j)$th cells. Therefore, Eq. (1) can be simplified as

$$a(i, j, t+1) = C_0 \oplus \begin{pmatrix} C_1 a(i+1, j, t) \oplus C_2 a(i, j-1, t) \\ \oplus C_3 a(i, j, t) \oplus C_4 a(i, j+1, t) \oplus C_5 a(i-1, j, t) \end{pmatrix}. \quad (2)$$

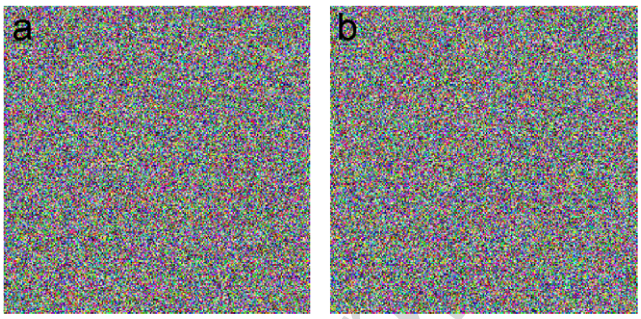

Fig. 1. Examples of CA data sequence, which are generated from the two-state/8 × 8-cell *von Neumann* 2-D hybrid CA.

Because Eq. (2) can be implemented by XOR gates, it is referred to as a 1-bit *Programmable additive CA* (PACA). The desired $N \times N$-bit CA [27–29] can be built for various applications using the 1-bit 2-D von Neumann PACA structure. Consequently, various combinations of rules, initial configurations and boundary conditions were imposed on a two-state/$N \times N$-cell CA to generate the states of the CA. The generalized CA data sequence is a sequence of pseudorandom numbers with high randomness under control. The methodology of Ref. [30] was extended to produce sequences of pseudorandom numbers using a two-state/$N \times N$-cell *von Neumann* 2-D hybrid CA in an image encryption application. Fig. 1 presents two examples of 24576 × 8 8-bit generalized CA data sequences which were generated from two-state/8 × 8-cell *von Neumann* 2-D hybrid CA with uniform initial state $6C_{16}$ with uniform "1" boundaries and a cyclic upper left-most corner.

3. Proposed image encryption/decryption method

The basic idea of the proposed image encryption/decryption method is to change the pixel values. The pixel values are changed by data reformation and CA substitution. Data are reformed using data reformation keys, while CA substitution is performed using CA keys. This section firstly describes the CA encryption/decryption scheme and then presents the proposed image encryption/decryption method.

3.1. CA encryption/decryption scheme

The CA keys with $(6+n) + N^2 + 4N$ bits are required to assign a specific CA evolution. In CA keys, the first $(6+n) = (6 + \lceil 2\log_2(N) \rceil)$ bits are rule-controlled data that specify CA rule numbers, according to the size of CA, where $\lceil x \rceil$ rounds the elements of x to the nearest integers $\geqslant x$; the next N^2 bits are initial data that specify initial configurations; finally, the remaining $4N$ bits specify the boundary conditions. Since $T \times N$ N-bit generalized CA data have $(T \times N)!$ possible permutations, extra bits are required to specify a particular permutation. However, since $(T \times N)!$ may be a huge number,

$n_W = \log_2(T \times N) = \log_2 T + \log_2 N = n_T + n_N$ bits are used to represent and generate a specific permutation, called a *linear permutation*. Notably, n_T bits and n_W bits are used to specify the time step and location, respectively, of the starting point for retrieving the $T \times N$ N-bit generalized CA data to generate a new sequence of N-bit CA data for encryption and decryption. In summary, the length of the CA key is $((6+n)+N^2+4N+n_w)$ bits. The CA key is of variable length according to the size of the 2-D CA and the number of time steps. Fig. 2 shows the structure of the CA key. CA keys are used to control the CA generating scheme (the dashed line block in Fig. 3) to generate a sequence of N-bit CA data for CA substitution. The CA-encrypted substitution is recursive to change the pixel value of the image for CA encryption. Fig. 3 shows the CA encryption/decryption scheme. In CA encryption, the signal of encryption/decryption control is set to 1; simultaneously, the input is a sequence of N-bit data and the output of recursive CA-encrypted substitution is a sequence of N-bit encrypted data.

Let $F(i), 0 \leqslant i \leqslant L_1 - 1$ be a sequence of N-bit input data and $CA_p(i), 0 \leqslant i \leqslant L_1 - 1$ be a sequence of N-bit CA data. Then, the recursive CA-encrypted substitution is defined as

$$CA\ encryption:$$
$$\begin{cases} E(0) = F(0), \\ E(i) = [F(i) + GCAT(E(i-1), CA_p(i))] \bmod 2^N, \\ \quad 1 \leqslant i \leqslant L_1 - 1. \end{cases} \tag{3}$$

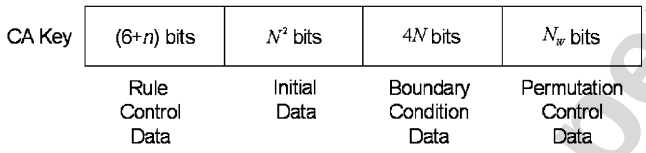

Fig. 2. Structure of CA key.

$GCAT(E(i-1), CA_p(i))$ means that $E(i-1)$ and $CA_p(i)$ execute the *generalized CA transform*. The general form of GCAT is expressed as

$$GCAT(E(i-1), CA_p(i)) = \begin{cases} ((E(i-1) + L_S) \oplus CA_p(i)) \bmod 2^N, \\ \overline{((E(i-1) + L_S) \oplus CA_p(i))} \bmod 2^N, \end{cases}$$

where $0 \leqslant L_S \leqslant 2^N - 1$ are values of level shift. 2^{N+1} GCATs exist of which the following six types were used in the simulation:

$$Type\ 1: EXOR(E(i-1), CA_p(i)) = E(i-1) \oplus CA_p(i), \tag{4}$$

$$Type\ 2: NEXOR(E(i-1), CA_p(i)) = \overline{E(i-1) \oplus CA_p(i)}, \tag{5}$$

$$Type\ 3: ALU\_1(E(i-1), CA_p(i)) = ((E(i-1) + 128) \oplus CA_p(i)) \bmod 2^N, \tag{6}$$

$$Type\ 4: ALU\_2(E(i-1), CA_p(i)) = \overline{((E(i-1) + 128) \oplus CA_p(i))} \bmod 2^N, \tag{7}$$

$$Type\ 5: ALU\_3(E(i-1), CA_p(i)) = ((E(i-1) + 1) \oplus CA_p(i)) \bmod 2^N, \tag{8}$$

$$Type\ 6: ALU\_4(E(i-1), CA_p(i)) = \overline{((E(i-1) + 1) \oplus CA_p(i))} \bmod 2^N. \tag{9}$$

Types 1 and 2 are the logic exclusive OR and the non-exclusive OR operations, respectively; both are with $L_S = 0$.

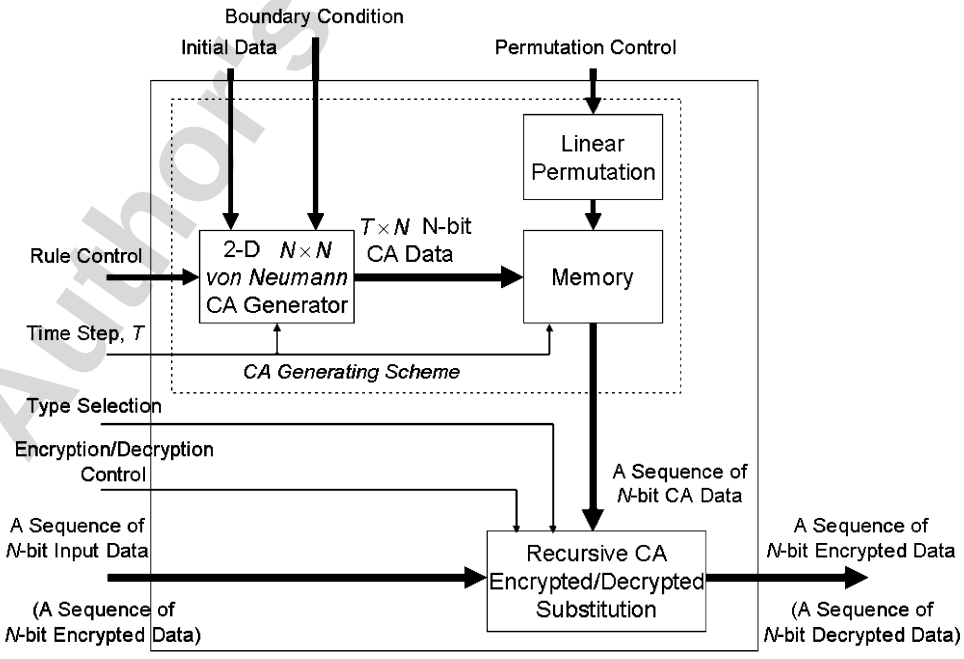

Fig. 3. CA encryption/decryption.

Types 3, 4, 5 and 6 are combinations of arithmetic and logical operations. $L_S = 128$ was used to change the most significant bit of $E(i-1)$ in Types 3 and 4, while, $L_S = 1$ was used to change the least significant bits (1-bit or $\geqslant 2$ bits) of $E(i-1)$ in Types 5 and 6. Since only six types of GCAT were used in the simulation, 3-bit *Type Selection* data were used to specify the type of GCAT.

Every secure encryption method must exhibit two fundamental properties. The first is the confusion property, which requires that encrypted data have an approximately random appearance, that is, pixel values are uniformly distributed. The second is the diffusion property with respect to original data and keys, which requires that similar original data generate completely different encrypted data when encrypted with the same key, and similar keys generate completely different encrypted data when used to encrypt the same original data. The properties of confusion and diffusion are detailed elsewhere [15,21,31]. The proposed recursive CA-encrypted substitution exhibits both confusion and diffusion properties. The confusion and diffusion properties are achieved by transforming the sequence $F(i), 0 \leqslant i \leqslant L_1 - 1$ into the sequence $E(i), 0 \leqslant i \leqslant L_1 - 1$ according to Eq. (3). The sequence $E(i), 0 \leqslant i \leqslant L_1 - 1$ yields uniformly distributed pixels because the high-quality random key-stream $CA_p(i), 0 \leqslant i \leqslant L_1 - 1$ is used in the transformation.

Reversing the operations of recursive CA encryption is to perform recursive CA decryption. In Fig. 3, the signal of encryption/decryption control is set to zero, causing CA decryption. Let $E(i), 0 \leqslant i \leqslant L_1 - 1$ be a sequence of N-bit encrypted data. Then, the recursive CA-decrypted substitution is given by

CA decryption:

$$\begin{cases} D(0) = E(0), \\ D(i) = [E(i) - GCAT(E(i-1), CA_p(i))] \bmod 2^N, \\ \quad 1 \leqslant i \leqslant L_1 - 1. \end{cases} \tag{10}$$

The $GCAT(E(i-1), CA_p(i))$ for recursive CA-decrypted substitution can be one of the types 1, 2, 3, 4, 5 and 6 as defined in Eqs. (4)–(9), respectively. Notably, the generalized CA transforms $GCAT(E(i-1), CA_p(i))$ for encryption and for decryption are identical. Fig. 4 shows the block diagram of recursive CA-encrypted/decrypted substitution. When encryption/decryption control is set to one, recursive CA encryption substitution is performed. When encryption/decryption control is set to zero, recursive CA decryption substitution is performed. Since the CA encryption/decryption scheme is lossless, the sequence of N-bit decrypted data $D(i), 0 \leqslant i \leqslant L_1 - 1$ is identical to the original sequence $F(i), 0 \leqslant i \leqslant L_1 - 1$. In Fig. 4, block z^{-1} executes a one-clock time delay because $E(i-1)$ is a one-clock delayed version of $E(i)$.

3.2. CA-based image encryption/decryption

Fig. 5 presents the CA-based image encryption/decryption system. The secret keys for encryption and decryption consist of four components—the data reformation key, the GCAT-type selection key, the CA key and the iteration key. These keys are identical and are known to both the sender and the receiver before the encrypted image is communicated. The data reformation key has two bits that are used to reformat the data of the image sequence $A(k,l), 0 \leqslant k,l \leqslant N' - 1$ as 4-bit (00_2), 8-bit (01_2), 16-bit (11_2) or 32-bit (10_2). The 3-bit GCAT-type selection key is used to select a particular type of GCAT. The CA key is used for a chosen CA rule number, initial data, boundary conditions and linear permutations to generate a CA key-stream for CA substitution. The iteration key is used to repeat the encryption process with specified times to obtain a more random encrypted image.

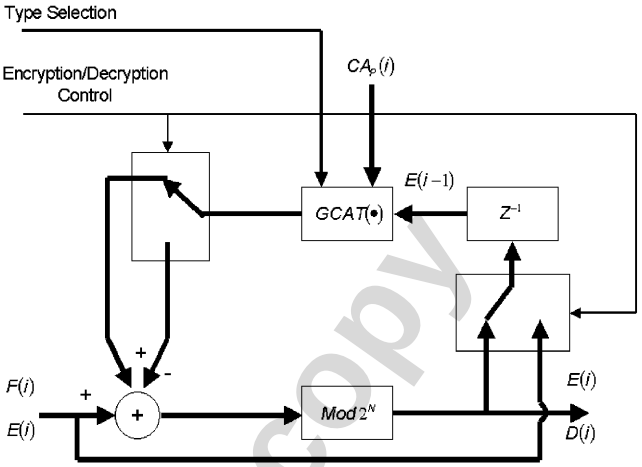

Fig. 4. Recursive CA encrypted/decrypted substitution.

A single pixel change in input image $A(k,l), 0 \leqslant k,l \leqslant N' - 1$ causes all the data in $E(i), 0 \leqslant i \leqslant L_1 - 1$ to be changed as follows. On the sender side, suppose a single pixel is changed in $A(k,l), 0 \leqslant k,l \leqslant N' - 1$. Then, the corresponding pixel at location j in $F(i), 0 \leqslant i \leqslant L_1 - 1$ is changed. After data reformation and CA encrypted substitution, all data between j and $L_1 - 1$ in $E(i)$ are changed. A small change in the data reformation key, or the CA key also changes all of the data in $E(i), 0 \leqslant i \leqslant L_1 - 1$, because a change in these two keys causes a large change in $F(i), 0 \leqslant i \leqslant L_1 - 1$, and/or the output sequence $E(i), 0 \leqslant i \leqslant L_1 - 1$.

On the receiver side, a data reformation key, a type selection key, a CA key, an iteration key, and a sequence of N-bit encrypted data are required. The data will be decrypted as follows. First, the recursive CA-decrypted substitution performs CA decryption to generate the sequence of N-bit decrypted data $D(i), 0 \leqslant i \leqslant L_1 - 1$. Then, an inverse data reformation produces the $N' \times N'$ decrypted image. The receiver should repeat these two procedures until the number of iterations is reached. Notably, the final $N' \times N'$ decrypted image is the same as the $N' \times N'$ original image because the proposed image encryption method is lossless.

4. Simulation results

Many simulations were conducted to illustrate various characteristics of the proposed CA-based image encryption/

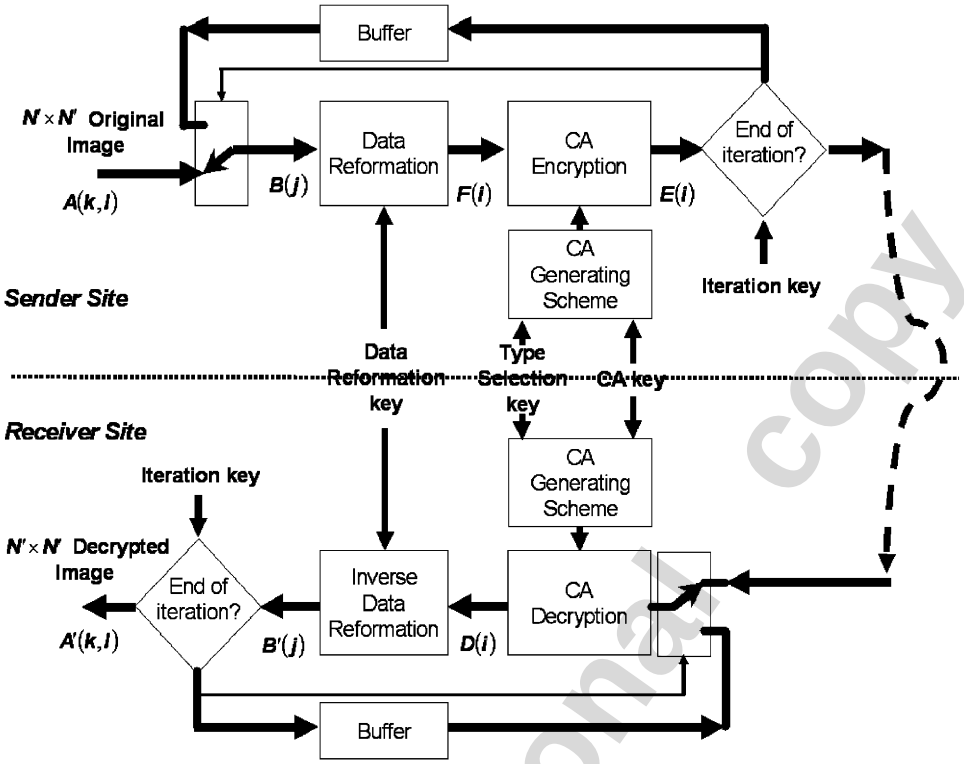

Fig. 5. CA-based image encryption/decryption system.

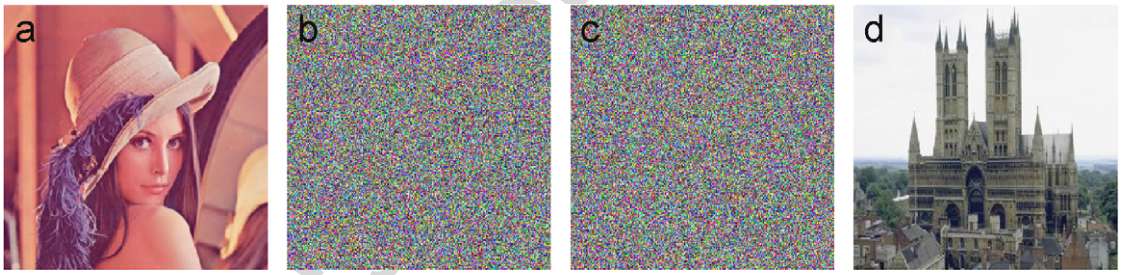

Fig. 6. Color test images and corresponding encryptions: (a) Original Lena, (b) Lena encrypted using the key-stream in Fig. 1a, (c) Lena encrypted using the key-stream in Fig. 1b, (d) original Lincoln Tower: the corresponding encryptions are similar to those in (b) and (c).

decryption system including pixel rearrangement, confusion and diffusion. Notably, all images in this section were 256 × 256. Fig. 6a and d show two color images Lena and Lincoln Tower, which were used to evaluate the performance of the proposed CA-based image security system. The data reformation key was 01_2, meaning the data for encryption and decryption were 8-bit, and that 2-D 8 × 8-cell *von Neumann* CA was adopted. The type selection key was 00_2, meaning that type 1 GCAT was applied to perform the recursive CA-encrypted and CA-decrypted substitutions. The CA key-streams were as presented in Fig. 1a and b. Fig. 6b and c present the corresponding encrypted Lena images; the corresponding encrypted Lincoln Tower images are similar to those in Fig. 6b and c because the proposed CA-based image security system achieves an excellent encryption.

In the following, some gray-level images were used to demonstrate the confusion and diffusion properties of the proposed security system. For simplification, the number of iterations was set to one. Two CA keys were used, which were CA key 1—uniform initial states $6C_{16}$, with "0" boundaries and cyclic boundary in the lower right-most corner, and CA key 2—which is obtained by 1-bit change to the initial data of CA key 1 the initial state of the upper left-most cell was changed from 0 to 1. The (6+6)-bit rule control data were 011111000001_2, such that the 2-D 8 × 8-cell dual-state *von Neumann* CA evolution was controlled by the rule number assignment 011111_2, based on the specified hybrid CA set data 000001_2. After the initial data, the boundary condition data and the rule control data had been determined, the 2-D 8 × 8-cell dual state *von Neumann* hybrid CA ran over 8192 time steps

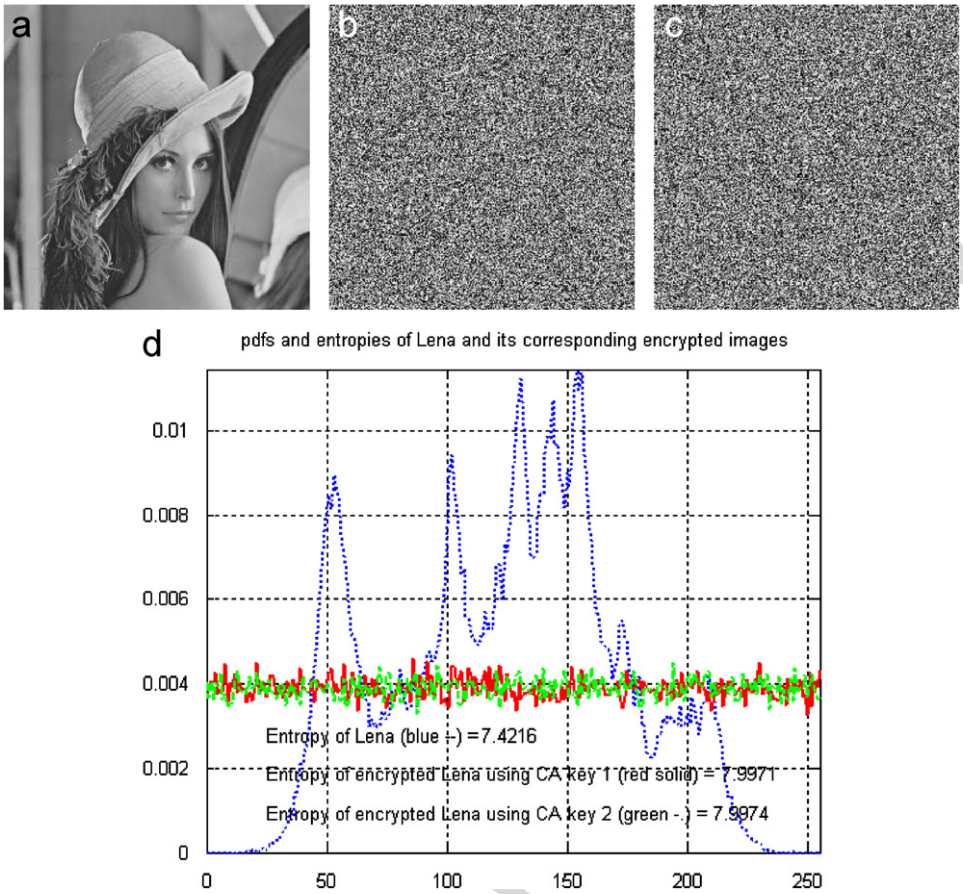

Fig. 7. Lena image proving the confusion property: (a) Original image, (b) image encrypted using CA key 1, (c) image encrypted using CA key 2, and (d) pdfs and entropies.

to generate 8192 × 8 8-bit generalized CA data. Then, 8-bit permutation control data 00000000_2 guided the system to perform a linear permutation from the first byte of the generalized CA data to generate a high-quality random sequence as a key-stream. Fig. 7a shows a Lena image and its encrypted images; Fig. 7b plots the corresponding probability density functions (pdfs) and entropies. These encrypted images were decrypted to produce the decrypted images, which were exactly identical to the original Lena image. This fact further demonstrated that the proposed CA-based image security system is effective for gray images.

4.1. Confusion property

The pdfs in Fig. 7 reveal that the encrypted Lena images comprised uniformly distributed pixels. Therefore, the proposed CA-based image security system exhibited the confusion property. The Jet image (Fig. 8a) and a pure black image (Fig. 9a) were encrypted using the same data reformation key (01_2), CA keys and Type 1 GCAT as that of Lena image to illustrate further the confusion property of the proposed system. Figs. 8b and c, and 9b and c show the encrypted images of the Jet and the pure black image, respectively. These encrypted images had almost uniform pdfs as presented in Figs. 8d and 9d, regardless of the original images, exhibiting the confusion property of the proposed system.

4.2. Diffusion property

The proposed system exhibits the diffusion property, meaning that any small change to the original image or secret key causes a significantly different output. The Lena image was modified by changing the value of one randomly chosen pixel by 1-bit to determine the diffusion property of the proposed system. The value of pixel (0, 0) was changed from 162 to 93. Both the original Lena and the modified Lena were encrypted using the same secret keys as before. Fig. 10 shows the scaled difference between these two encrypted images; the figure reveals that the two encrypted images exhibited no similarities, even though their original images differed by only one pixel, illustrating the diffusion property of the proposed system for the images.

As mentioned above, different CA keys produce different key-streams, even though the difference is quite small. For example, the initial data of CA key 2 is a small change of CA key 1 in the upper left-most corner (the initial state is changed from 0 to 1), but significantly different key-streams are produced. Finally, these different key-streams result in vastly different

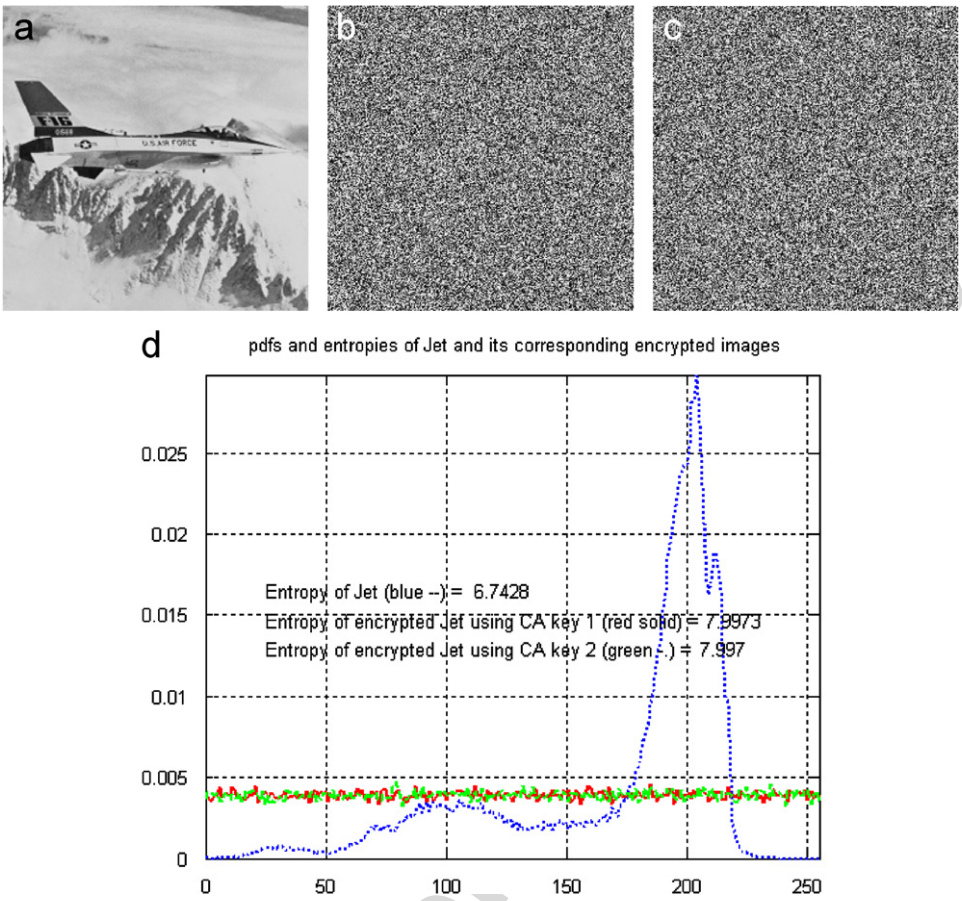

Fig. 8. Jet image proving the confusion property: (a) Original image, (b) image encrypted using CA key 1, (c) image encrypted using CA key 2, and (d) pdfs and entropies.

encrypted images, as presented in Figs. 7b and c, 8b and c, and 9b and c. Therefore, the diffusion property of the proposed CA-based image encryption/decryption method with respect to CA keys is illustrated in detail.

The data reformation key has two bits that are used to reformat the data type of an input image as N-bit, where N can be $4(00_2)$, $8(01_2)$, $16(11_2)$ or $32\ (10_2)$. Furthermore, the 2-D *von Neumann* CA generator determines how many CA cells $(N \times N)$ will work, according to the data reformation key. Simulations show that a small change in the data reformation key, such as $00_2 \rightarrow 01_2 \rightarrow 11_2 \rightarrow 10_2$ is a 1-bit change, and can produce significantly different sequences of CA data, resulting in very different encrypted images. Fig. 11 presents the scaled difference between two encrypted images which were generated using different data reformation keys 01_2 and 11_2. Therefore, the proposed CA-based image encryption/decryption system has the diffusion property with respect to data reformation keys.

5. Possible secret keys, cryptanalysis and comparisons

As discussed previously, a 2-D $N \times N$-cell dual-state *von Neumann* hybrid CA that runs over T time steps produces $2^{32\times N^2+N^2+4N} \times (T \times N)!$ possible groups of $T \times N$ N-bit generalized CA data. However, for efficient computer simulation or logic-gate implementation, the 6-bit rule control data are used to specify specific CA rule numbers. Furthermore, since $T \times N$ N-bit generalized CA data have $(T \times N)!$ possible permutations, it could be a very large number. Therefore, the linear permutation with $n_W = \log_2(T \times N) = \log_2 T + \log_2 N = n_T + n_N$ bits is compact in representing and generating a specific set of permutations. In summary, in the computer simulation, $2^{(6\times N^2+N^2+4N+n_W)\times i}$ possible groups of $T \times N$ N-bit generalized CA data were used. Notably, the proposed system specifies four data types and six GCAT types. Thus, Table 1 presents a high volume of secret keys.

The CA-based image security system is a stream cipher, meaning that it uses a specified secret key to generate a key-stream to encrypt a plaintext string, according to Eq. (3). Therefore, the length of the key-stream must equal or exceed that of the plaintext to be secure. Furthermore, the length of a CA state cycle is very important in determining the effectiveness of CA as a generator of random numbers. As described elsewhere [30], the average cycle length for a 2-D $N \times N$-cell dual-state *von Neumann* hybrid CA increases exponentially and is of the order of 2^{N^2-3} for $N < 8$ or 2^{N^2-4} for $N \geqslant 8$. Therefore, a suitable 2-D $N \times N$-cell CA must be chosen to produce a high-quality key-stream for encryption, according to the size of the

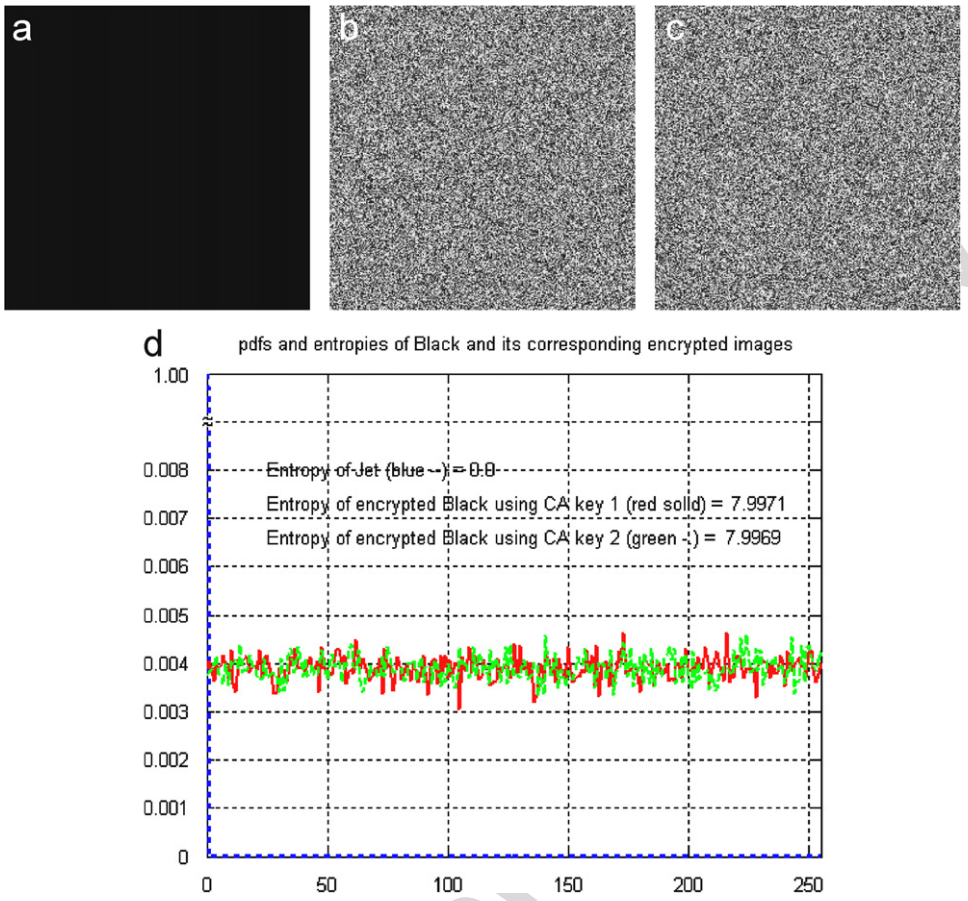

Fig. 9. Black image proving the confusion property; (a) Original image, (b) image encrypted using CA key 1, (c) image encrypted using CA key 2, and (d) pdfs and entropies.

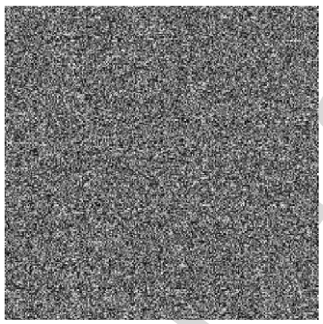

Fig. 10. Scaled difference between two encrypted images, indicating that the two encrypted images were not at all similar, even though their original images differed by only one pixel.

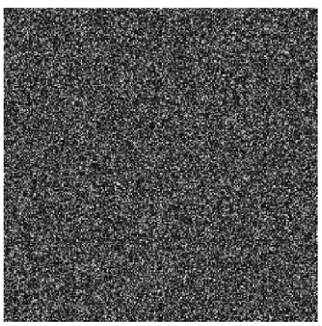

Fig. 11. Scaled difference between two encrypted images, indicating that the two encrypted images were not at all similar, even though their data reformation keys differed by only one bit (changing 01_2 to 11_2).

image. The relationship between the image size and the minimum size of a suitable 2-D CA is described as follows: if a $2^{n_1} \times 2^{n_2}$ image has to be encrypted, then the minimum size of a suitable 2-D CA is $\lceil\sqrt{n_1+n_2+3}\rceil \times \lceil\sqrt{n_1+n_2+3}\rceil$ for $\lceil\sqrt{n_1+n_2+3}\rceil < 8$ or $\lceil\sqrt{n_1+n_2+4}\rceil \times \lceil\sqrt{n_1+n_2+4}\rceil$ for $\lceil\sqrt{n_1+n_2+4}\rceil \geqslant 8$. For example, for some of the color and gray-level images with a size of $2^{n_1} \times 2^{n_2} = 256 \times 256$ used in the simulation, the minimum size of a suitable 2-D CA was 5×5. However, a 2-D dual-state *von Neumann* hybrid CA with a size of 8×8 was adopted to facilitate reconfigurable hardware implementation. Therefore, most cycle lengths of a 2-D 8×8-cell dual-state *von Neumann* hybrid CA were longer than 2^{60}, producing an 8-bit key-stream with a cycle length of over 2^{63} larger than the tested images which had a length of 2^{16}.

Security systems have to withstand all attacks such as ciphertext only, known plaintext, chosen plaintext and chosen ciphertext. The ability of the proposed image security system to withstand some of these attacks is considered below.

Table 1
Possible secret key for herein simulation

Iteration keys	Possible data reformation keys	Possible GCAT-type selection	2-D $N \times N$ *von Neumann* CA	Time steps T	Possible CA keys $2^{6\times N^2+N^2+4N+n_W}$	Possible secret keys
i	4	6	4×4	32768	2^{114}	$> 10^{35\times i}$
			8×8	8192	2^{496}	$> 10^{150\times i}$
			16×16	2048	2^{1871}	$> 10^{562\times i}$
			32×32	512	2^{7310}	$> 10^{2194\times i}$

Table 2
List of comparisons

Algorithm	CA-based image security system ($N \times N = 8 \times 8$, one iteration)	DCPcrypt		
		RC4	Triple-DES	AES
Classification	Stream cipher	Stream cipher	Block cipher	Block cipher
Key length (bits)	121	256	192	256
Complexity of cryptanalysis	$> 2^{496}$	$2^{30.6}$ [33]	2^{112} [34]	2^{224} [35]
CPU encryption time[a] (ms/3 Mbytes)	31 ± 16	70 ± 8	609 ± 16	180 ± 8
CPU decryption time[a] (ms/3 Mbytes)	31 ± 16	70 ± 8	609 ± 16	180 ± 8
Entropy of ciphertext (bits)	7.9999	7.9999	7.9999	7.9999

[a]CPU encryption/decryption time is the CPU processing time for encrypting/decrypting 3 Mb of plaintext/ciphertext excluding the time required for hard disk storage.

The CA-based image security system was defined as a tuple $(\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{K}, \boldsymbol{L}, \boldsymbol{E}, \boldsymbol{D})$, together with g functions that are expressed in Eqs. (3)–(10) and satisfied the following conditions: (1) P was a finite set of possible plaintexts; (2) C was a finite set of possible ciphertexts; (3) K, the key-space, was a finite set of $2^{(6\times N^2+N^2+N+n_W)\times i}$ possible secret keys; (4) L was a finite set called the key-stream; (5) g was the key-stream generator, g took the CA key, a partial part of secret key $K \in \boldsymbol{K}$, as input, and generated an infinite string $\mathbf{CA}_p = CA_p(0)CA_p(1)\cdots$ called the key-stream, where $CA_p(i) \in \boldsymbol{L}\ \forall i$; (6) for each $CA_p(i) \in \boldsymbol{L}$, an encryption rule $e_{CA_p(i)} \in \boldsymbol{E}$ applied and the corresponding decryption rule was $d_{CA_p(i)} \in \boldsymbol{D}$, where $e_{CA_p(i)} : \boldsymbol{P} \to \boldsymbol{C}$ and $d_{CA_p(i)} : \boldsymbol{C} \to \boldsymbol{P}$ were functions such that $d_{CA_p(i)}(e_{CA_p(i)}(F(i))) = F(i)$ for every plaintext element $F(i) \in \boldsymbol{P}$. Suppose that the probability distribution on the plaintext space is P. The plaintext element defines a random variable, $\mathbf{F}$. The *a priori* probability is $p[\mathbf{F} = F(i)]$. The key-stream is also defined as a random variable, $\mathbf{CA}_p$. The probability associated with key-stream $CA_p(i)$ is selected as $p[\mathbf{CA}_p = CA_p(i)]$. The key-stream is generated using a specified CA key, which is randomly selected by legal users before they know what the plaintext will be. Therefore, the reasonable assumption is made that the key-stream and the plaintext are independent random variables. The two probability distributions on P and K induce a probability distribution on C. Thus, the ciphertext element can be considered to be a random variable, $\mathbf{E}$. From the probability $p[\mathbf{E} = E(i)]$, $E(i)$ can be easily computed as the transmitted ciphertext. For a CA key $K \in \boldsymbol{K}$ and the corresponding key-stream, define $E(i) = C(F(i)) = e_{CA_p=g(k)}(F(i)) : F(i) \in \boldsymbol{P}$. Then, for every $E(i) \in \boldsymbol{C}$, $p[\mathbf{E} = E(i)] = p[\mathbf{CA}_p = CA_p(i)]p[\mathbf{F} = d_{CA_p}(E(i))]$. For any $E(i) \in \boldsymbol{C}$ and $F(i) \in \boldsymbol{P}$, the conditional probability $p[\mathbf{E} = E(i) \mid \mathbf{F} = F(i)], \forall i$ can be computed as $p[\mathbf{E} = E(i) \mid \mathbf{F} = F(i)] = p[\mathbf{CA}_p = CA_p(i)], \forall i$. The conditional probability $p[\mathbf{F} = F(i) \mid \mathbf{E} = E(i)], \forall i$ can now be determined using Bayes' theorem as $p[\mathbf{F}=F(i) \mid \mathbf{E}=E(i)] = p[\mathbf{F} = F(i)]p[\mathbf{CA}_p = CA_p(i)]/p[\mathbf{E} = E(i)], \forall i$. From Eq. (3), one $CA_p(i) \in \boldsymbol{L}$ is specified to encrypt $F(i) \in \boldsymbol{P}, \forall i$ as $E(i) \in \boldsymbol{C}$, that is $p[\mathbf{CA}_p = CA_p(i)] = P[\mathbf{E} = E(i)] = 1/L_1, \forall i$. Thus, this image encryption method meets the perfect secrecy condition $p[\mathbf{F} = F(i) | \mathbf{E} = E(i)] = p[\mathbf{F} = F(i)], \forall i$.

The cryptanalyst can yield no information about the plaintext by observing the ciphertext because of the system's perfect secrecy. This result proves that the system can withstand ciphertext only, and chosen ciphertext attacks. The cryptanalyst can use known plaintext and chosen plaintext attacks to this scheme to obtain CA keys because the cryptanalyst cannot obtain information about the plaintext by observing the ciphertext. However, known plaintext and chosen plaintext attacks on this scheme are more difficult than those on encrypted data streams because the system has many possible CA keys.

The software implementation of the proposed CA-based image security system was performed using an Intel® P4 CPU (3.2 GHz) personal computer with Microsoft® Windows® XP and Borland® C++ builder® 5.0. The performance of the proposed image security system is now compared using DCPcrypt [32] with that of three published algorithms—RC4, Triple-DES and AES (Rijndael). DCPcrypt is OSI-certified open source software, containing 29 cryptographic components for

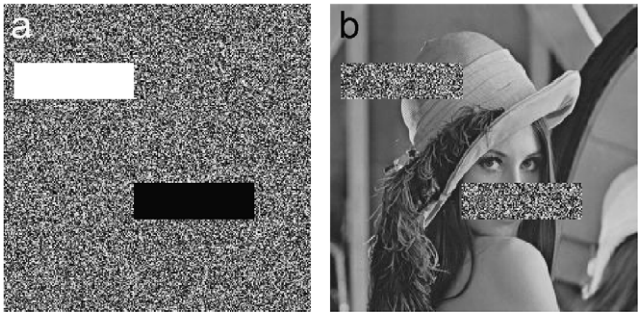

Fig. 12. Survival property of proposed system: (a) Corrupted ciphertext (encrypted image), (b) decryption of corrupted ciphertext.

programming languages Delphi®, C++ Builder® and Kylix®. Of these three DCPcrypt cryptographic components, RC4 [21] is the most widely used stream cipher in the software with a key length of 256 bits; Triple-DES is a block cipher and a variant of DES [21], with a block size of 64 bits and a key length of 192-bit; AES [23] is the successor of DES with a block size of 128 bits with a key of length 256 bits. Three color image files (Lena, Lincoln Tower, and Jet) of 3 Mb are used for comparison. Table 2 summarizes the results, indicating that the performance of the proposed system is superior to that of RC4, Triple-DES and AES, since it has shorter keys; involves more complicated cryptanalysis and has shorter CPU decryption and decryption time.

The proposed CA-based image security system is also a synchronous cipher because the key-stream of the proposed system is generated independently of the plaintext and the ciphertext, making it susceptible to synchronization problems. In a synchronous stream cipher, the sender and receiver must be in step for decryption to be successful. If digits are added or removed from the image during transmission, then synchronization is lost. One approach to solve the synchronization problem is to tag the ciphertext with markers at regular points in the output. However, if a small fraction of the ciphertext is corrupted in transmission, rather than added or lost, then only the corresponding fraction in the plaintext is affected and the error does not propagate to other parts of the plaintext; a large area of the image therefore survives. This characteristic makes the proposed system reliable in transmissions with high error rate. Fig. 12 shows the survival properties of the proposed system.

6. Conclusions

This work presents a novel image security system based on 2-D *von Neumann* CA. The encryption method is based on the replacement of the pixel values using a recursive CA substitution. The advantages of the proposed CA-based image security system can be summarized as follows: (1) Secret keys are the data reformation key, the type selection key, the CA key and the iteration key, which are of variable lengths producing a large number of possible secret keys, more than $10^{35\times i}$–$10^{2194\times i}$—according to the size of the 2-D *von Neumann* CA and the number of iterations. (2) Choosing a suitable size for the 2-D CA, according to the size of the image, enables the system to withstand the cropping-and-replacement attack. (3) The system is economic in consuming computational resources because the encryption/decryption scheme uses integer arithmetic and logic operations. Comparative results show that the performance of the proposed system is superior to those of RC4, Triple-DES, and AES because of its shorter keys, more complicated cryptanalysis, and shorter CPU decryption and decryption time for a particular ciphertext entropy. This system withstands the survival against attack as well as RC4, Triple-DES, and AES.

Acknowledgments

The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this research under Contract No. NSC-93-2215-E-239-003. The anonymous reviewers are appreciated for their valuable comments.

References

[1] N.G. Bourbakis, C. Alexopoulos, Picture data encryption using SCAN patterns, Pattern Recognition 25 (6) (1992) 567–581.
[2] C. Alexopoulos, N.G. Bourbakis, N. Ioannou, Image encryption method using a class of fractals, J. Electron. Imaging (4) (1995) 251–259.
[3] N.G. Bourbakis, Image data compression encryption using G-SCAN patterns, in: Proceedings of IEEE Conference on SMC, Orlando, Florida, USA, October 1997, pp. 1117–1120.
[4] S.S. Maniccam, N.G. Bourbakis, Image and video encryption using SCAN patterns, Pattern Recognition 37 (4) (2004) 725–737.
[5] J. Scharinger, Fast encryption of image data using chaotic Kolmogorov flows, J. Electron. Imaging 7 (2) (1998) 318–325.
[6] F. Pichler, J. Scharinger, Finite dimensional generalized Baker dynamical system for cryptographic application, Lecture Notes in Computer Science, vol. 1030, Springer, Berlin, 1995, pp. 465–476.
[7] S.J. Li, G.R. Chen, X. Zheng, Chaos-based encryption for digital image and videos, in: B. Furht, D. Kirovski (Eds.), The Multimedia Security Handbook, CRC Press LLC, Boca Raton, FL, October 2004(Chapter 4).
[8] K.L. Chung, L.C. Chang, Large encrypting binary images with higher security, Pattern Recognition Lett. 19 (5) (1998) 461–468.
[9] X.B. Li, J. Knipe, H. Cheng, Image compression and encryption using tree structures, Pattern Recognition Lett. 18 (11) (1997) 1253–1259.
[10] K.C. Chang, J.L. Liu, A linear quadtree compression scheme for image encryption, Signal Process. Image Commun. 10 (4) (1997) 279–290.
[11] T.J. Chuang, J.C. Lin, New approach to image encryption, J. Electron. Imaging 7 (2) (1998) 350–356.
[12] X.L. Wu, P.W. Moo, Joint image/video compression and encryption via high order conditional entropy coding of wavelet coefficients, in: Proceedings of IEEE International Conference on Multimedia Computing and Systems, 1999, pp. 908–912.
[13] T.J. Chuang, J.C. Lin, A new multiresolutional approach to still image encryption, Pattern Recognition Image Anal. 9 (3) (1999) 431–436.
[14] C.J. Kuo, Novel image encryption technique and its application in progressive transmission, J. Electron. Imaging 2 (4) (1993) 345–351.
[15] S.S. Maniccam, N.G. Bourbakis, Lossless image compression and encryption using SCAN, Pattern Recognition 34 (6) (2001) 1229–1245.
[16] I. Karafyllidis, I. Andreadis, P. Tzionas, Ph. Tsalides, A. Thanailakis, A cellular automaton for the determination of the mean velocity of moving objects and its VLSI implementation, Pattern Recognition 29 (4) (1996) 689–699.
[17] S. Nandi, B.K. Kar, P. Pal Chaudhuri, Theory and applications of cellular automata in cryptography, IEEE Trans. Comput. 43 (12) (1994) 1346–1357.

[18] O. Lafe, Data compression and encryption using cellular automata transform, Eng. Appl. Artif. Intell. 10 (6) (1998) 581–591.
[19] K. Sasidhar, S. Chattopadhyay, P. Pal Chaudhuri, CAA decoder for cellular automata based error correcting code, IEEE Trans. Comput. 45 (9) (1996) 1003–1016.
[20] P. Hortensius, R. McLeod, W. Pries, M. Miller, H. Card, Cellular automata-based pseudorandom number generators for built-in self-test, IEEE Trans. Comput. Aided Des. Integrated Circuits Syst. 8 (8) (1989) 842–859.
[21] B. Schneier, Applied Cryptography, Wiley, New York, 1996.
[22] G. Brassard, Modern Cryptology, Springer, New York, 1988.
[23] S. Landau, Standing the test of time: the data encryption standard, in: Notices of American Mathematical Society, March 2000, pp. 341–349.
[24] S.R. Blackburn, S. Murphy, K.G. Paterson, Comments on the theory and applications of cellular automata in cryptography, IEEE Trans. Comput. 2 (5) (1997) 637–638.
[25] N.G. Bourbakis, C. Alexopoulos, A. Klinger, A parallel implementation of the SCAN language, Int. J. Comput. Lang. 14 (4) (1989) 239–254.
[26] S. Wolfram, Statistical mechanics of cellular automata, Rev. Mod. Phys. 55 (3) (1983) 601–644.
[27] R.J. Chen, J.L. Lai, VLSI implementation of the universal one-dimensional CAT/ICAT, in: Proceedings of the 2002 IEEE Asia-Pacific Conference on Circuit and Systems (APCCAS'02), vol. 2, Bali, Indonesia, October 28–31, 2002, pp. 279–282.
[28] R.J. Chen, J.L. Lai, Y.T. Lai, Design of the universal 2-D cellular automata bases generator and its VLSI implementation, in: Proceedings of the Seventh World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2003), vol. XII, Orlando, Florida, USA, July 27–31, 2003, pp. 165–168.
[29] R.J. Chen, J.L. Lai, C.S. Yang, W.C. Fan, W.J. Chen, C.C. Hung, L.Y. Hsu, The architecture of the re-configurable 2-D cellular automata bases generator, in: Proceedings of the 14th VLSI Design/CAD Symposium (VLSI Design/CAD 2003), Hualien, Taiwan, August 12–15, 2003, pp. 137–140.
[30] M. Tomassini, M. Sipper, M. Perrenoud, On the generation of high-quality random numbers by two-dimensional cellular automata, IEEE Trans. Comput. 49 (10) (2000) 1146–1151.
[31] C.E. Shaannon, Communication theory of secrecy systems, Bell Syst. Tech. J. 28 (4) (1949) 656–715.
[32] D. Barton, DCPcrypt cryptographic component library v2 beta 3, ⟨http://www.cityinthesky.co.uk/cryptography.html⟩, 1999.
[33] S.R. Fluhrer, D.A. McGrew, Statistical analysis of the alleged RC4 keystream generator, Lecture Notes in Computer Science, vol. 2259, Springer, Berlin, 2001, pp. 1–24.
[34] S. Lucks, Attacking triple encryption, Lecture Notes in Computer Science, vol. 1372, Springer, Berlin, 1988, pp. 239–253.
[35] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, D. Whiting, Improved cryptanalysis of Rijndael, Lecture Notes in Computer Science, vol. 1978, Springer, Berlin, 2000, pp. 213–230.

About the Author—RONG-JIAN CHEN (M'02, SM'04 IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 1987, 1991, and 1995, respectively. He joined the faculty of the National Lien-Ho Institute of Technology in August 1995, where he was the Chair of the Department of Electronic Engineering during the 1999–2001 academic years. At present, he is Associate Professor of the Department of Electronic Engineering, National United University. He is a member of Nano-electronics and Giga-scale System Technical Committee, IEEE CAS Society. He serves as Co-Editor of Globalization Leadership Column at IEEE Circuits and Devices Magazine (per invitation from EIC Dr. Ron Waynant), and serves on the Editorial Board of IEEE Circuits and Systems Magazine (per invitation from EIC Prof. Maciej Ogorzalek). He also serves as Associate Editor of IEEE Trans. on Circuits and Systems, Part 1 from December 15, 2005. His research interests include digital image/video processing, neural networks, and VLSI design of multimedia system.

About the Author—JUI-LIN LAI (M'01, SM'05 IEEE) received the B.S. degree from the Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, in 1984. He received the M.S.E.E. and Ph.D. degree in the Institute of Control Engineering and the Institute of Electronic Engineering from the National Chiao-Tung University, Taiwan, in 1990 and 2004, respectively. At present, he is Associate Professor of the Department of Electronic Engineering, National United University. His research interests include analog and digital VLSI design, neural networks, and computing architecture, and nanotechnology.