

Data Encryption Using Non-uniform 2-D Von Neumann Cellular Automata

Rong-Jian Chen and Jui-Lin Lai

Department of Electronics Engineering, National United University

No. 1 Lien-Da, Kung-Ching Li, Miao-Li, Taiwan 360, ROC

rjchen@nuu.edu.tw, jllai@nuu.edu.tw

Abstract— This paper presents a new method for data encryption. Its encryption scheme is based on replacement of the data values. The data values are replaced using a progressive cellular automata (CA) substitution. In the progressive CA substitution, the key stream is generated from the non-uniform 2-D $N \times N$ von Neumann cellular automata, that is a special type of discrete cellular neural networks (CNN). The characteristics of the proposed encryption method are loss-less, symmetric private key encryption, very large number of security keys, key-dependent permutation, and key-dependent pixel value replacement. Simulation results for color images show that the proposed data encryption method satisfies the properties of confusion and diffusion due to the CA substitution is wonderful.

Key Words—discrete cellular neural networks, non-uniform 2-D von Neumann cellular automata, data encryption and decryption

I. INTRODUCTION

With the ever-increasing growth of multimedia applications, security is an important issue in communication and storage of images, and encryption is one the ways to ensure security. Image encryption has applications in inter-net communication, multimedia systems, medical imaging, telemedicine, and military communication. There already exist several data and/or image encryption methods. They include IDEA method [1] and RSA method [2] for data encryption; SCAN-based methods [3], chaos-based methods [4], tree structure-based methods [5], and other miscellaneous methods [6] for image encryption; [7, 8] for encryption of compressed images. However, each of them has its strength and weakness in terms of security level, speed, and resulting stream size metrics. The proposed encryption method is based on replacement of the pixel values. The data values are replaced using a progressive CA substitution with a key stream that is generated from the CA evolution rules. The proposed encrypted system is loss-less, key-dependent permutation, and key-dependent pixel value replacement. Additionally, it is a symmetric private key security, meaning that the same key is needed for encryption and decryption; both sender and receiver must know the key.

Reasons that we used CA for data encryption are described as follows. (a) CA has been applied successfully to several physical systems, processes, and scientific problems where local interactions are involved, such as image processing [9], data encryption [9], byte error correcting code [10], and as pseudorandom number generators for VLSI built-in self-test [11]. (b) The number of CA evolution rule is very large; we hence have very large number of ways to produce a sequence of CA data for image encryption and decryption. (c) Progressive CA substitution is integer arithmetic and/or logic operation.

which is an easy and simple computation.

This paper is organized as follows. Section II describes the proposed encryption method. Simulation results are drawn in Section III. Finally, section IV gives the discussions and conclusions.

II. THE PROPOSED ENCRYPTION METHOD

A. CA Encryption/Decryption Scheme

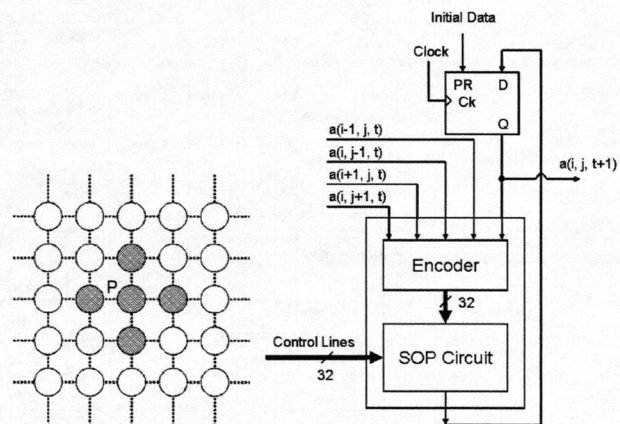


Fig. 1 2-D von Neumann CA, (a) 2-D von Neumann CA space, (b) the structure of 1-bit PCA

In 2-D CA space, the specified node P, with its four nearest neighbors form the von Neumann neighborhood. Fig. 1a shows the 2-D von Neumann CA space. The state of the given node at time step (t+1) will be determined from the states of nodes within its neighborhood at time step t. Using a specified rule, the states are updated synchronously in time steps for all cells. Let $a(i, j, t)$ represent the state of (i, j) th cell at time t, whose von Neumann neighborhoods are in the states: $a(i-1, j, t)$, $a(i, j-1, t)$, $a(i+1, j, t)$, and $a(i, j+1, t)$. Then the rule of 2-D von Neumann CA evolution way can be expressed as

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

where F is a Boolean function that defines the rule.

The hardware implementation of Equation (1) for 1-bit 2-D von Neumann CA is shown in Fig. 1b. Such a structure is referred as a *programmable* CA (PCA). Using the 1-bit 2-D von Neumann PCA structure, one can build the desired non-uniform $N \times N$ -bit cellular automata. Due to the non-uniform 2-D $N \times N$ -bit cellular automata uses $4N + 32N^2 + N^2$ input pins to set boundary condition, rule control, and initial data, it shall cost many input pads, we therefore use three separated memories to reduce the input pads. The architecture of non-uniform 2-D $N \times N$ von Neumann CA

generator for hardware implementation is shown in Fig. 2. Given a $N \times N$ -cell dual-state von Neumann 2-D CA runs over T time steps, it has $2^{2^2} = 2^{32}$ rules, $2^{N \times N}$ initial configurations, 2^{4N} boundary conditions, and results in $2^{32N^2 + N^2 + 4N}$ CA evolution ways for generating $T \times N$ N-bit generalized CA data. Consequently, cyclic boundary conditions were imposed on a 2-state/3-site/ $N \times N$ -cells CA to generate the states of the automata. The CA generating scheme shown in dash-line block of Fig. 3, which is controlled by CA key to

generate a sequence of N-bit CA data $CA_p(i)$, $0 \leq i \leq L_1 - 1$ for CA substitution. In order to change the data values to achieve CA encryption, the CA encrypted substitution is progressive. The scheme of CA encryption/decryption scheme is shown in Fig. 3. For CA encryption, the pin of encryption/decryption control is set to be 1; at the same time, the input is a sequence of N-bit data and the output of progressive CA encrypted substitution is a sequence of N-bit encrypted data.

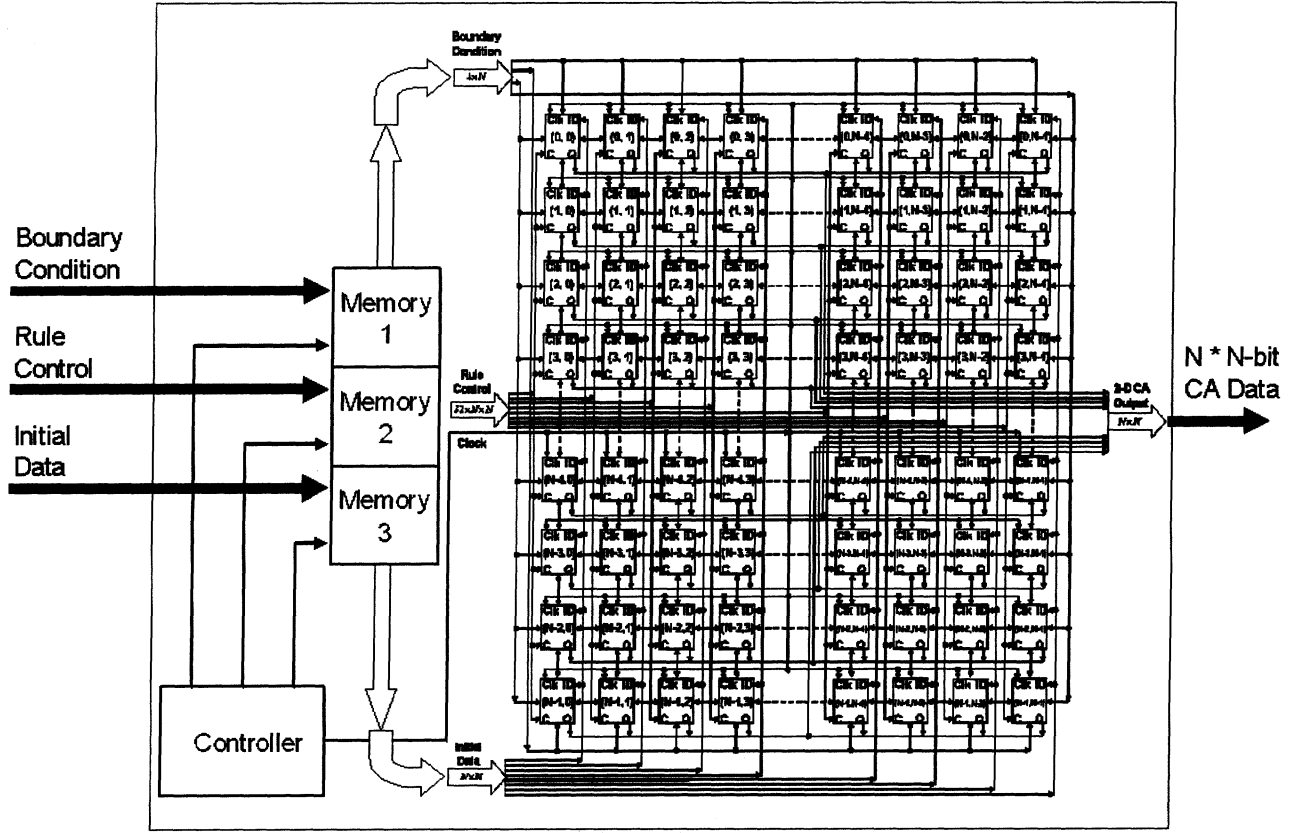


Fig. 2 The architecture of non-uniform 2-D $N \times N$ CA generator

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

CA encryption:

$$\begin{cases} E(0) = F(0) \\ E(i) = [F(i) + GCAT(E(i-1), CA_p(i))] \bmod 2^N, 1 \leq i \leq L_1 - 1 \end{cases} \quad (2)$$

The $GCAT(E(i-1), CA_p(i))$ means that $E(i-1)$ and $CA_p(i)$ execute the generalized CA transform. $GCAT(E(i-1), CA_p(i))$ can be expressed as

$$\text{Type 1: } EXOR(E(i-1), CA_p(i)) = E(i-1) \oplus CA_p(i), \quad (3)$$

$$\text{Type 2: } NEXOR(E(i-1), CA_p(i)) = \overline{E(i-1) \oplus CA_p(i)}, \quad (4)$$

$$\text{Type 3: } ARITH_1(E(i-1), CA_p(i)) = (E(i-1) \times CA_p(i)) \bmod 2^N, \quad (5)$$

$$\text{Type 4: } ARITH_2(E(i-1), CA_p(i)) = ((E(i-1) + 1) \times CA_p(i)) \bmod 2^N, \quad (6)$$

$$\text{Type 5: } ALU_1(E(i-1), CA_p(i)) = ((E(i-1) + 1) \oplus CA_p(i)) \bmod 2^N, \quad (7)$$

Type 6:

$$ALU_2(E(i-1), CA_p(i)) = (\overline{E(i-1) + 1} \oplus CA_p(i)) \bmod 2^N. \quad (8)$$

Type 1 and 2 are the logic exclusive OR and the not exclusive OR operation respectively. Type 3 and 4 are arithmetic operations. As for type 5 and 6, they are the combination of the arithmetic and the logic operations. We use Type Selection Bus in Fig. 2 to decide the type of $GCAT(E(i-1), CA_p(i))$.

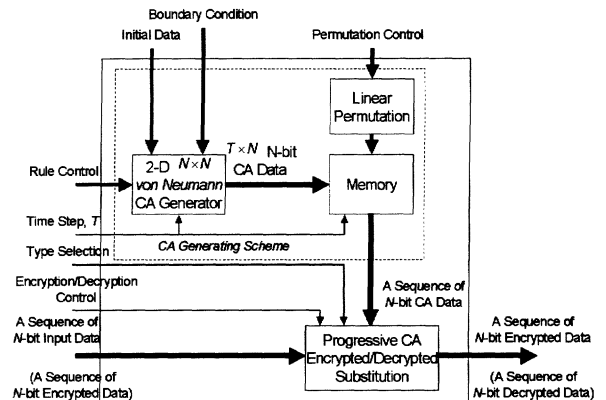


Fig.3 Scheme of CA encryption/decryption

The proposed progressive CA encrypted substitution satisfies

both confusion and diffusion properties. The confusion and diffusion properties are achieved by transforming the sequence $F(i), 0 \leq i \leq L_1 - 1$ into the sequence $E(i), 0 \leq i \leq L_1 - 1$ using Equation (2). The sequence $E(i), 0 \leq i \leq L_1 - 1$ gets uniformly distributed pixels because the pseudo random sequence $CA_p(i), 0 \leq i \leq L_1 - 1$ is used in the transformation. Therefore, the sequence $E(i), 0 \leq i \leq L_1 - 1$ gets confusion property. The sequence $E(i), 0 \leq i \leq L_1 - 1$ gets diffusion property because a single change in value $F(i)$ changes $E(i)$ which changes $E(i+1)$ which changes $E(i+2)$ and changes propagate up to the end of the sequence.

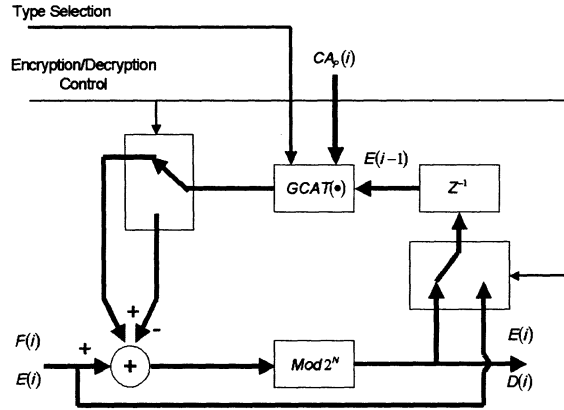


Fig. 4 Block diagram of the progressive CA encrypted/decrypted substitution

The progressive CA decryption is the reversing operation of progressive CA encryption. In Fig. 4, the pin of encryption/decryption control is set to be 0, which causes the scheme of CA encryption/decryption to perform the progressive CA decryption. Let $E(i), 0 \leq i \leq L_1 - 1$ be a sequence of N -bit encrypted data. Then the progressive CA decrypted substitution can be expressed as

CA decryption:

$$\begin{cases} D(0) = E(0) \\ D(i) = [E(i) - GCMAT(E(i-1), CA_p(i))] \bmod 2^N, 1 \leq i \leq L_1 - 1 \end{cases} \quad (9)$$

In CA decryption, the $GCMAT(E(i-1), CA_p(i))$ is identical to that for encryption. The block diagram of the progressive CA encryption/decryption substitution is shown in Fig. 4. When encryption/decryption control is set to be 1, it performs progressive CA encryption substitution. Whereas, encryption/decryption control is set to be 0, it works progressive CA decryption substitution. Due to the CA encryption/decryption scheme is loss-less, the sequence of N -bit decrypted data $D(i), 0 \leq i \leq L_1 - 1$ is the identification of the original sequence $F(i), 0 \leq i \leq L_1 - 1$.

B. The proposed Encryption System

The proposed encryption system is shown in Fig. 5. The security keys for encryption and decryption consist of three components, namely, iteration key, type selection key and CA key. These keys are identical and are known to both the sender and the receiver before the communication of encrypted data. Iteration key is used for repeating encryption process a specified times to get more random encrypted data. Type selection key is

used for selecting the type of $GCMAT(E(i-1), CA_p(i))$ to perform progressive CA encryption/decryption substitution. Whereas, the CA key is used for deciding CA rule number, initial data, boundary conditions, and linear permutation to generate a sequence of CA data for CA substitution.

In the sender site, suppose it has iteration key, type selection key, CA key, and input data. The CA encryption replaces the data values to produce a sequence of N -bit encrypted data $E(i), 0 \leq i \leq L_1 - 1$. These processes are repeated until the specified iterations are finished. The receiver site performs inverse operation of the sender site. Suppose receiver has received iteration key, type selection key, CA key, and a sequence of N -bit encrypted data. The progressive CA decrypted substitution performs CA decryption to generate the sequence of N -bit decrypted data $D(i), 0 \leq i \leq L_1 - 1$. These processes are repeated a specified iterations to produce the N' decrypted image.

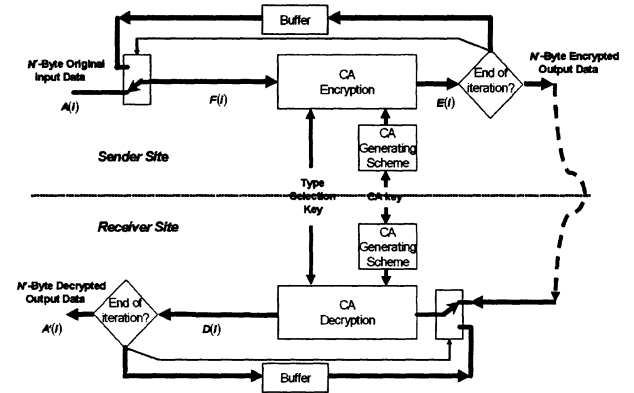


Fig. 5 The proposed encryption system

III. SIMULATION RESULTS

The proposed encryption system performed well encryption not only the general text data but also compressed images and uncompressed images. Several simulations were conducted to test various properties of the proposed image encryption system that include confusion and diffusion properties. Note that in all the following experiments, all images are of size 256×256 . Fig. 6a shows a YUV formatted color Lena image that is used for testing the performance of the proposed image encryption system. The data type for encryption and decryption is 8-bit, and non-uniform 2-D 8×8 -cell von Neumann CA is selected. The CA key consist of the rule control data $0FFEFF2F_{16}$, $6C_{16}$ uniform initial states, zero boundaries with cyclic boundary at right down corner, and linear permutation with 00_{16} . It is clear that the rule control data is $0FFEFF2F_{16}$, which means that the 2-D 8×8 -cell dual-state von Neumann CA evolution is controlled by a specified set of functions that was stored in memory 2. Once the initial data, boundary condition data, and rule control data were decided, the 2-D von Neumann CA run over 8192 time steps to generate the generalized CA data of size 65536. Then 8-bit permutation control data 00_{16} guides the system to do linear permutation from the first 8-bit data of the CA initial state (1st time step) to generate the pseudo random sequence of CA data. We used type 1 GCMAT to perform the progressive CA encryption and decryption substitutions. The iteration is set to 1, for the reason

of simplification. Encrypted images of Lena are shown in Fig. 6b. Y-signal histograms of the Lena image (blue line) and the encrypted Lena image (green line) were shown in Fig. 7. It shows that the encrypted Lena image gets uniformly distributed pixels. This fact illustrates that the proposed encryption system satisfies the confusion property. This encrypted image performs the process of decryption to produce the decrypted image. The decrypted image is exactly identical to the original Lena image. This fact shows that the proposed image encryption system works well as our expectation.

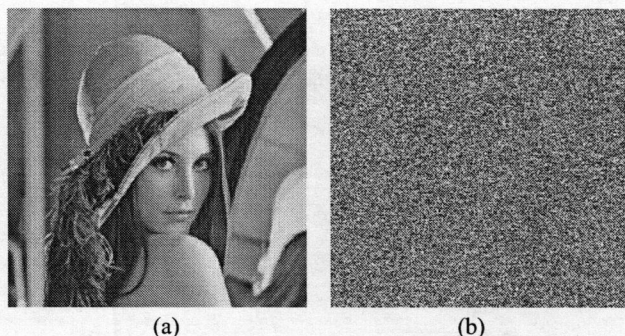


Fig. 6 (a) Test image (Lena), (b) Encrypted Lena image

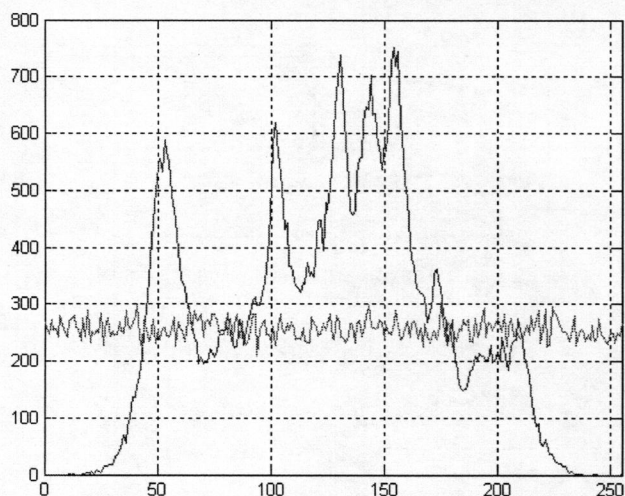


Fig.7 Y-signal histogram of the original and the encrypted Lena image, blue line shows the original, whereas green line shows the encrypted.

In order to determine the diffusion property of the proposed system with respect to images, the Y-signal of Lena image was modified by incrementing the value of one randomly chosen pixel by 1. The value of pixel (0, 0) was incremented from 161 to 162. Both the original Lena and the modified Lena were encrypted using the same secret keys. The pixel-wise absolute difference of two encrypted images is displayed in Fig. 7, which shows that the two encrypted images have no similarities even though their original images differ by only one pixel. Thus, it proves the diffusion property of the proposed system with respect to images.

IV. DISCUSSIONS AND CONCLUSIONS

As previous mention, we have possible $2^{32N^2+N^2+4N}$ groups of $T \times N$ N-bit CA data in our simulation. Thus, for compressed images are of size N' 8-bit bytes, T

becomes $8 \times N' / N^2$, it has a high volume of security keys with the order of $\left[10^{852} \times \left(\frac{8N'}{N^2} \right)! \right]^{iter}$.

This paper presented a new encryption system based on the non-uniform 2-D von Neumann CA. The encryption method is based on replacement of the data values. The data values are replaced using a progressive CA substitution. We summarize the characteristics of the proposed encryption system are:

- Loss-less encryption of data.
- Symmetric security system.
- Security key consists of iteration key, type selection key, and CA key, which is variable length with huge number of possible of security keys.
- Confusion and diffusion properties are satisfied. Almost perfect guess of encryption key makes decryption impossible.
- Encryption/ decryption scheme uses integer arithmetic and logic operations, it can be easily hardware implemented.

ACKNOWLEDGEMENTS

The National Science Council of the Republic of China under the contract NSC-92-2218-E-239-003 supported this work.

REFERENCES

- X. Lai, "On the design and security of block ciphers," Berlin, Germany: Hartung-Gorre Verlag, 1992, vol. 1.
- R Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120-126, 1978.
- N. Bourbakis, C. Alexopoulos, "Picture data encryption using SCAN patterns," *Pattern Recognition*, vol. 25, no. 6, pp. 567-581, 1992.
- J. Scharinger, "Fast encryption of image data using chaotic Kolmogorov flows," *Electronic Imaging*, vol. 17, no. 2, pp. 318-325, 1998.
- L. Chang, "Large encrypting of binary images with higher security," *Pattern Recognition Letter*, vol. 19, no. 5, pp. 461-468, 1998.
- T. Chuang and J. Lin, "New approach to image encryption," *Electronic Imaging*, no. 4, pp. 350-356, 1998.
- H. K.-C. Chang and J.-L. Liu, "A linear quadtree compression scheme for image encryption," *Signal Process.: Image Commu.*, vol. 10, no. 4, pp. 279-290, Sep. 1997.
- D. Jones, "Application of splay trees to data compression," *Commun. ACM*, pp. 996-1007, Aug. 1988.
- Olu Lafe, "Data compression and encryption using Cellular Automata transform," *Eng. Applic. Artif. Intell.*, vol. 10, no. 6, pp. 581-591, 1998.
- K. Sasidhar, S. Chattopadhyay, and P. Pal Chaudhuri, "CA decoder for cellular automata based error correcting code," *IEEE Trans. On Computers*, vol. 45, pp. 1003-1016, 1996.
- P. Hortensius, R. McLeod, W. Pries, M. Miller, and H. Card, "Cellular Automata-based pseudorandom number generators for built-in self-test," *IEEE Trans. On Computers Aided Design*, vol. 8, pp. 842-859, 1989.