An Application of Keystream Using Cellular Automata for Image Encryption in IoT

*Shyi-Tsong Wu**

Department of Electronic Engineering, National Ilan University, Taiwan
stwu@niu.edu.tw

Abstract

Recently, there have been some studies on security applications using cellular automata. A cellular automaton (CA) is characterized by simplicity and high-speed computation, making it suitable for the resource-constrained IoT environment. In this paper, we first merge a sliding-window, bit permutation, and a CA and propose a keystream generator that has the inherent CA advantages of simplicity and speed. The sliding-window provides the unpredictability of the keystream and gains strength in terms of security. The produced keystream has passed the tests of NIST SP800-22. On the basis of the proposed sliding-window CA keystream generator, an image encryption system is further proposed. We use the keystream generated to scramble and confuse the original image. The histogram of the encrypted image is more average. It can resist differential attacks, and in the differential attack analysis, the *NPCR* and *UACI* values are at least 99% and 33%, respectively.

Keywords: Cellular automata, Internet of Things, Sliding-window, Image encryption, Keystream generator

1 Introduction

For security, the current cryptosystem generally requires complex operations. However, under the resource-constrained Internet of Things (IoT), a complex system reduces communication efficiency [1-2]. The cellular automaton (CA) is a suitable method in this regard [3-4]. A cellular automaton (CA) is characterized by simplicity and high-speed computation and suitable for the application in IoT. Currently cellular automata are used in the field of cryptography. Many encryption systems based on cellular automata have been proposed [5-9]. A CA can serve as a source of random numbers to be used for encryption messages and other applications [10]. To increase the unpredictability of secure applications, different CA rules are merged on the basis of simple logic operations of the CA [9, 11].

The proposed keystream generator is based on iteration. The output data of one iteration are the input for the keystream generator for the next iteration of the CA [12-13]. In this paper, we introduce the sliding-window concept to strengthen the keystream generator in terms of security. A sliding-window makes the output data of one iteration, which will be the input for the next iteration, variable, making the output keystream unpredictable. This can further increase the variability of the input data for the third iteration and make the output more chaotic. The proposed keystream generator combining a sliding-window, a CA, and bit permutation is suitable for resource-constrained IoT systems. The produced keystream can resist typical attacks and has passed the tests of NIST SP800-22 [14].

Research on image encryption has garnered a lot of attention [15-22]. The key-stream generated by the proposed sliding-window cellular automaton (SWCA) keystream generator is applied to disturb and confuse an image in order to encrypt it. In the security analysis, compared with histograms of images encrypted by traditional methods, the histogram of the image encrypted by this method is more average, which means that the distribution of image pixels is more uniform and the key space is larger. In the differential attack analysis, the Number of Changing Pixel Rate (*NPCR*) value and the Unified Averaged Changed Intensity (*UACI*) value are more than 99% and 33%, respectively. This means that the proposed image encryption system effectively resists differential attacks.

The CA has the merits of simple logic operations and high speed. In this paper, the application of CA to the security of IoT is our motivation. The contribution of the paper is the sliding-window of CA is utilized to promote the chaos of input data for each iteration of the keystream generator. Besides, to gain the security, we merged multiple CA in the proposed scheme. Finally, based on the generated keystream, an image scheme is built by scrambling and confusion. It inherent the advantage of CA and is suitable to IoT.

The paper is organized as follows: Section 2 introduces the CA. Section 3 depicts our proposed SWCA-based keystream generator and image encryption based on the proposed keystream generator. Section 4 illustrates the implementation and its result analysis to test the functionality of our proposed scheme. Security analysis and experimental results are also shown. Section 5 provides the conclusion.

2 Cellular Automata

2.1 Principle of CA

Keystream generators play a pivotal role in IoT communication security. The CA-based keystream generator

*Corresponding Author: Shyi-Tsong Wu; E-mail: stwu@niu.edu.tw
DOI: 10.53106/160792642023012401014

is important for data communication between various devices [23]. Neumann proposed the concept of a CA based on simple logic operation, which has the characteristics of simplicity and nonlinearity [10]. Various results are generated according to its own state, the state of its neighbors, and different rules of cellular automata. For the CA in a one-dimensional space, an iterative output consists of a logical value of 0 or 1. Mathematically, it can be defined as Equation (1):

$$a_{t+1}(i) = f(a_t(i-1), a_t(i), a_t(i+1)) \,, \tag{1}$$

where $a_t(i)$ represents the state of a bit, i represents the position of the bit in the sequence, t is the current state of the sequence, and $a_{t+1}(i)$ is the next state of the sequence. When the operation of one bit is iterated to the next bit, the logic operation will be performed according to the cell itself and the states of the neighborhoods. As shown in Figure 1, the bits on the left and on the right generate the bit state of the next iteration according to different CA rules. Cellular automata have different numbers of rules as per the different logical operations.

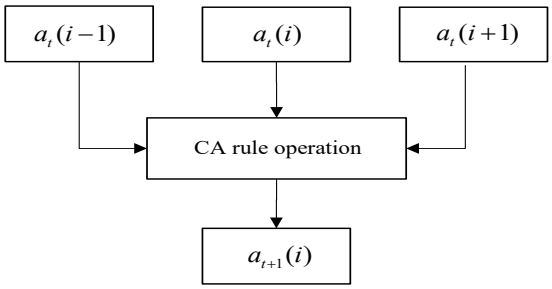


Figure 1. CA operation

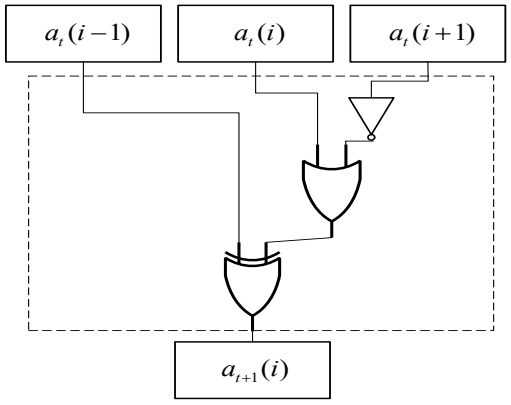
For various logical operations, different CA rules are formulated. Wolfram researched the CA in one-dimensional space and put forward 256 kinds of CA rules [24]. Table 1 shows some logical operations of different CA rules, which generate different output sequences according to different rules and input data. Simplification, regularization, and blockization characterize the keystream generated by the CA, and the CA is easy to implement on various platforms [4, 6, 25]. Therefore, the CA is an efficient method to generate keystreams that satisfy the characteristics of keystream generators in today's communications. For example, the operation of CA rule 45 has the logic operation shown in Equation (2).

$$a_{t+1}(i) = a_t(i-1) \oplus (a_t(i) \cup \overline{a_t(i+1)}) \,. \tag{2}$$

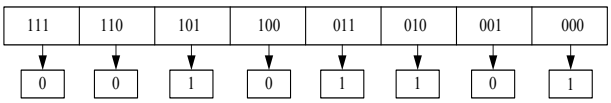
Table 1. The rules of cellular automata and their logical operation

CA rule	Logical operation
Rule 18	$(a_t(i-1) \oplus a_t(i) \oplus a_t(i+1)) \cap \overline{a_t(i)}$
Rule 30	$a_t(i-1) \oplus (a_t(i) \cup a_t(i+1))$
Rule 45	$a_t(i-1) \oplus (a_t(i) \cup \overline{a_t(i+1)})$
Rule 60	$a_t(i-1) \oplus a_t(i)$
Rule 86	$(a_t(i-1) \cup a_t(i)) \oplus a_t(i+1)$
Rule 90	$a_t(i-1) \oplus a_t(i+1)$
Rule 105	$a_t(i-1) \oplus a_t(i) \oplus \overline{a_t(i+1)}$
Rule 150	$a_t(i-1) \oplus a_t(i) \oplus a_t(i+1)$
Rule 165	$a_t(i-1) \oplus \overline{a_t(i+1)}$
Rule 182	$a_t(i-1) \cap a_t(i+1) \cup a_t(i-1) \oplus a_t(i) \oplus a_t(i+1)$

Figure 2 represents the input and output results of the cellular automaton rule 45 opera-tion. Figure 2(a) and Figure 2(b) are the logic circuit and truth table of CA rule 45, respectively. For two-state and one-dimension CA proposed by Stephen wolfram, the rule number given represents the decimal format the binary number encoding the rule table. For example, the CA rule 45 has the $f(1,1,1) = 0$, $f(1,1,0) = 0$, $f(1,0,1) = 1$, $f(1,0,0) = 0$, $f(0,1,1) = 1$, $f(0,1,0) = 1$, $f(0,0,1) = 0$, $f(0,0,0) = 1$. So it is denoted as CA rule 45, i.e. $(0010\ 1101)_b = 45$.



(a) Logic circuit of CA rule 45



(b) Truth table of CA rule 45

Figure 2. Logic circuit and truth table of CA rule 45

2.2 Related Work

The applications of cellular automata in security are widespread. Roy et al. proposed a symmetric block encryption system based on the CA [4]. This encryption system includes three main parts: the rule-vector generator, the encryption algorithm, and the decryption algorithm.

First, the rule-vector generator randomly generates GCA rule vectors that have cycle lengths of 8. They are the keys used for encryption and decryption. The proposed CA is claimed to be high in efficiency and lightweight in terms of operation load. However, in this study, the rule-vector generator is used as the key generator and the numbers of keys that satisfy the requirement of the rule-vector generator are rare. That means that the key space is smaller.

Kumar et al. proposed a block symmetric encryption method based on a lightweight two-dimensional CA [10]. The encryption process of a data block consists of an XOR operation, block permutation, one-dimensional reversible cellular automata keystream generation and two-dimensional reversible cellular automata Margolus neighborhood cell. The proposed lightweight symmetric key cryptography system can be applied to small IoT devices with low computational requirements.

Kumaravel et al. proposed a block encryption method [8]. The encryption is based on reversible cellular automata, non-uniform reversible cellular automata, and bit permutation (BP) to achieve higher parallelism and to simplify the hardware and software implementations for applications requiring a high degree of security. The message is encrypted by Pseudo Random Number Generators (PRNGs) using the CA, and it increases the complexity by mixing reversible cellular automata and non-reversible cellular automata. The symmetric block encryption is high-speed, simple, and effective.

CA is also adopted in one-way hash function. Rajeshwaran et al. proposed a CA-based hashing algorithm (CABHA) by using CA rules and a custom transformation function to create a strong hash from an input message and a key [9]. It can verify the integrity and authenticity of the message. The CA provides good nonlinear transformation, and the transformation function offers better diffusion and confusion, which ensures that the one-way hash function has a good avalanche effect. In this study, experiments have proved that this hash function has good collision resistance characteristics.

In IoT, image encryptions using cellular automata have been proposed [15-16, 18-19]. Roy et al. proposed an image encryption cipher called IESCA for IoT applications that use camera sensors for surveillance. The proposed cipher uses 2D Moore cellular automata with nine neighbors [26]. Two-dimensional Moore cellular automata cells can generate a highly chaotic sequence that can be used to encrypt images or generate pseudo random numbers. Since it is convenient to apply cellular automata in hardware and cellular automata consist of simple operations locally, the proposed scheme is lightweight and can be easily implemented in sensor devices at the physical level.

3 The Proposed Sliding-Window CA-Based Image Encryption

The proposed image encryption scheme is based on a keystream generated by the sliding-window CA (SWCA)-based keystream generator. In this section, we first introduce the SWCA-based keystream generator and then depict the image encryption scheme based on the proposed keystream.

3.1 The SWCA-based Keystream Generator

The simple logic operation of the CA can improve the efficiency of the overall secure communications in IoT. In this section, we propose an SWCA-based keystream generator. The initial states of the CA are iterated with neighboring cells in the process, and the output state exhibits diversity and unpredictability. It shows complex interactive phenomena under the relatively simple rules. The outputs of the states are suitable for keystream generators. Figure 3 is a block diagram of the proposed basic keystream generator, where the input key is 128 bit, r is the number of rounds, and keystream is the output.

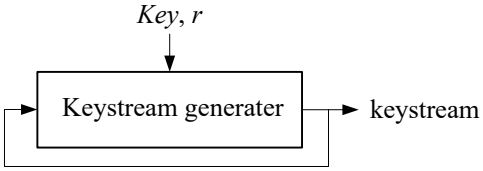


Figure 3. The proposed keystream generation process

The security is lower if the sequence is iterated using a single CA rule [12]. To increase the chaos of the output keystream and improve the randomness, we process the iteration by multiple CA rules. Figure 4 shows the composite use of multiple CA rules, which increases the chaos of the keystream. We further combine one-dimensional cellular automata, a sliding-window, and bit permutation to propose an SWCA-based keystream generator.

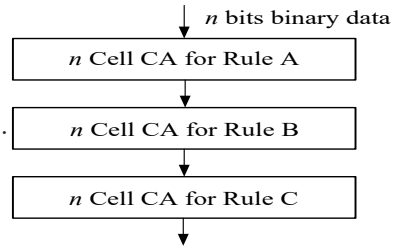


Figure 4. The combination of multiple CA rules

3.1.1 Sliding-Window

The output data of the first iteration are the input of the keystream generator of the second iterative operation of the CA. The sliding-window ensures that the range of the output data of the first iteration captured is variable. Therefore, the input data of the second iteration is variable, making the output more unpredictable. Before the CA operation, the sliding-window has to capture data. Figure 5 represents the operation flow of the sliding-window, where data a_{t-2} and a_{t-1} represent the output data of the previous two iterations, formulated as Equations (3) and (4).

$$a_{t-2} = \{a_{t-2}(0), a_{t-2}(1), a_{t-2}(2), \cdots a_{t-2}(127)\} . \quad \textbf{(3)}$$

$$a_{t-1} = \{a_{t-1}(0), a_{t-1}(1), a_{t-1}(2), \cdots a_{t-1}(127)\} . \quad \textbf{(4)}$$

The data a_{t-2} are concatenated with a_{t-1}, as shown in Equation (5). The sequence {b(0), b(1)...b(7)} determines the window displacement position j, as shown in Equation (6):

$$Row = a_{t-2} \,||\, a_{t-1} = \{b(0), b(1), b(2), \cdots b(255)\} . \quad \textbf{(5)}$$

$$j = b(0)\times 2^0 + b(1)\times 2^1 + b(2)\times 2^2 + b(3)\times 2^3 + b(4)\times 2^4 + b(5)\times 2^5 + b(6)\times 2^6 . \quad \textbf{(6)}$$

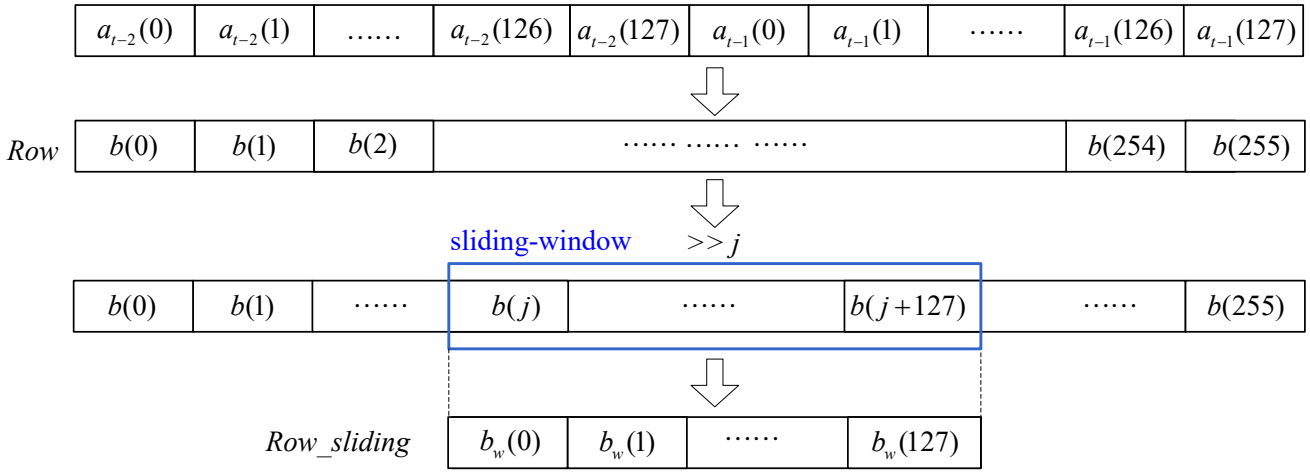


Figure 5. A sliding-window

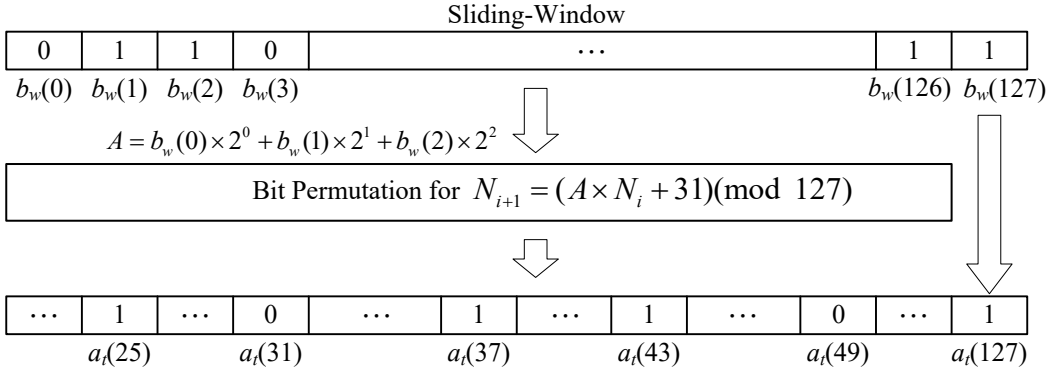


Figure 6. Bit permutation

The CA rule chooses the sequence to be performed on the basis of the starting position j. The starting position j also represents the starting position of the window. As shown in Equation (7), there are 128 bits in the window. The position of the window where the data are captured will be changed according to the number j, and it is similar to a sliding-window with displacement j.

$$Row\_sliding = \{b(j), b(j+1), \ldots b(j+127)\}. \quad \textbf{(7)}$$

For each SWCA iteration, a 128-bit sequence in the sliding-window is put to the next iteration of the CA operation. Because there are no neighbors on either side for CA operation, the output sequence will gradually shrink in length [1]. To obtain a 128-bit output constantly, we add 1 bit to the left and 1 bit to the right of the sliding-window. That is, $b_w(127)$ is added to the leftmost side and $b_w(0)$ to the rightmost side to form a 130-bit sequence {$b_w(127)$, $b_w(0)$, $b_w(1)\ldots b_w(127)$, $b_w(0)$}. After that, a one-dimensional cellular automaton operation is performed, which produces a 128-bit output sequence.

3.1.2 Bit Permutation

The bit permutation is basically performed by the linear congruence method (LCG) to change the positions of the bits in the sequence. For bit permutation, we use the linear congruence method, shown in Equation (8):

$$N_{i+1} = (A \times N_i + B)(\text{mod } M) . \quad \textbf{(8)}$$

Where A, B, M are parameters, N_i represents the original position, and N_{i+1} represents the replaced position. Figure 6 is an example of bit permutation. If $b_w(0)$, $b_w(1)$, $b_w(2)$ are equal to 011, i.e. A = 6, then the $b_w(0)$ bit is replaced to the position

of $a_t(31)$, and so on. The last 1 bit $a_t(127)$ is equal to $b_w(127)$. The purpose of the bit permutation is to break up the order of the entire 128-bit sequence. And the keystream generated by the cellular automaton is more unpredictable. For the consideration of efficiency, the relation of input and output can be looked up by table for actual operation.

The proposed keystream generator combines a one-dimensional CA, a sliding-window, and bit permutation to generate the keystream required for image encryption. The input data are the 128-bit key (*Key*), and the number of operation rounds are *r*. For each round, according to round counter, the data chosen via the sliding-window process a CA operation on the basis of the one-dimensional CA rules and bit replacement. Finally, a 128-bit sequence is obtained. The output data a_t replace the sequence a_{t-1}, and a_{t-2} is updated, i.e., the input candidates of the next iteration are renovated. The input value *r* determines the overall number of rounds, which affects the final output keystream length. Finally, an $r \times 128$ bit keystream is generated.

3.2 SWCA-based Image Encryption

In this section, we present an image encryption system based on the SWCA. With the vigorous development of digital information, people convert traditional photos into digital images. These images are often transmitted via the IoT. However, these images if not encrypted can be easily cracked or stolen during the transmission process, leading to violation of confidentiality and privacy. Therefore, in recent years, research on image encryption has garnered a lot of attention [15-16, 18-19, 27].

Ordinary images have obvious image characteristics. For example, among adjacent images, the same area of grayscale image data will have similar grayscale values. The high correlation among these images makes it easy for attackers to attack, and some traditional encryption systems may not provide image encryption that can withstand such attacks. Recently, many image encryption systems have been proposed. The image encryption process mainly involves scrambling columns and rows and confusing the pixels via the keystream. Scrambling changes the relative positions of the image pixels to encrypt the image. The computational complexity of the encryption process is low, and the efficiency is high. However, the image values of pixels are not changed. As a result, the final histogram is the same and attackers may attack using this weakness. Image confusion involves changing the values of the pixels using the keystream. It combines with image scrambling and promotes secure image encryption [28].

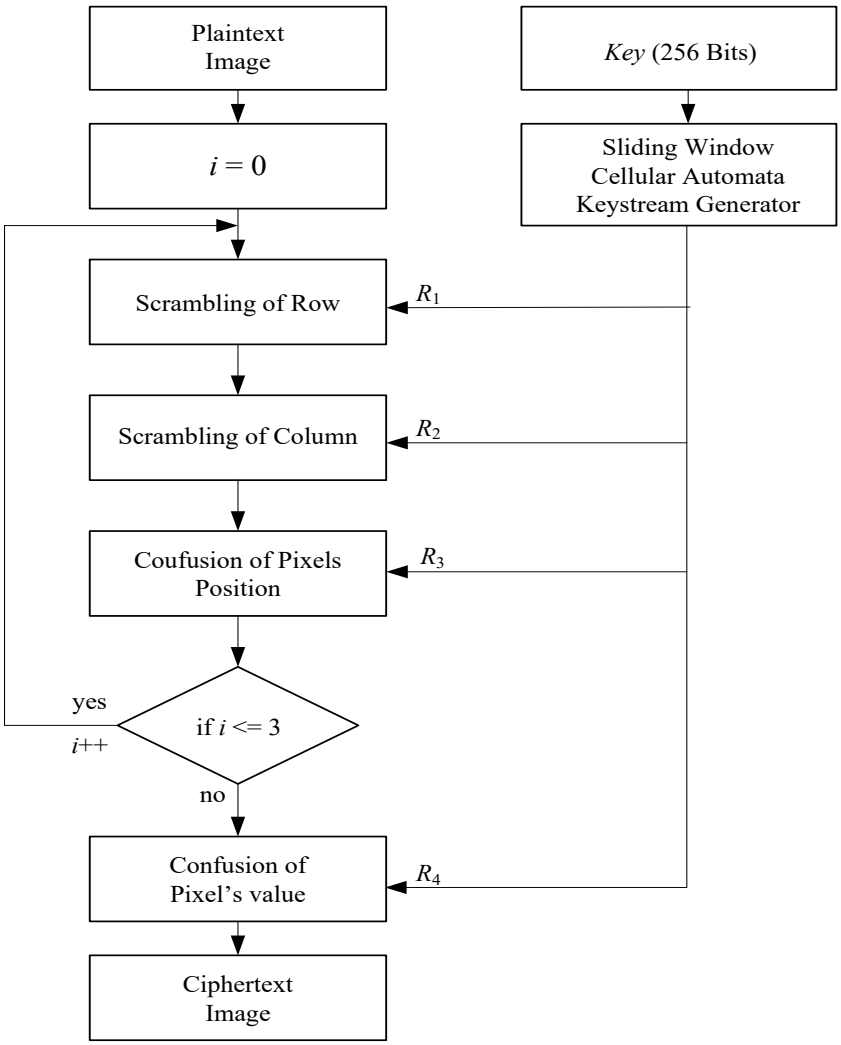


Figure 7. Flowchart of the proposed SWCA-based image encryption system

Since adjacent images have the same image characteristics, there will be security concerns when encrypting directly. To solve the problem of high correlation between adjacent images, we propose an image encryption system that combines image scrambling and image pixel location confusion based on the proposed SWCA keystream generator. Figure 7 displays this algorithm. First, the key *Key* is input into the keystream generator to generate the keystreams R_1, R_2, R_3, and R_4, required for each image encryption process. Then, the system repeats four rounds of row scrambling and column scrambling and pixel position confusion and performs pixel value confusion. The final output ciphertext is the encrypted image. The operations of the image encryption will be introduced in detail the following sections.

3.2.1 Scrambling of Rows and Columns

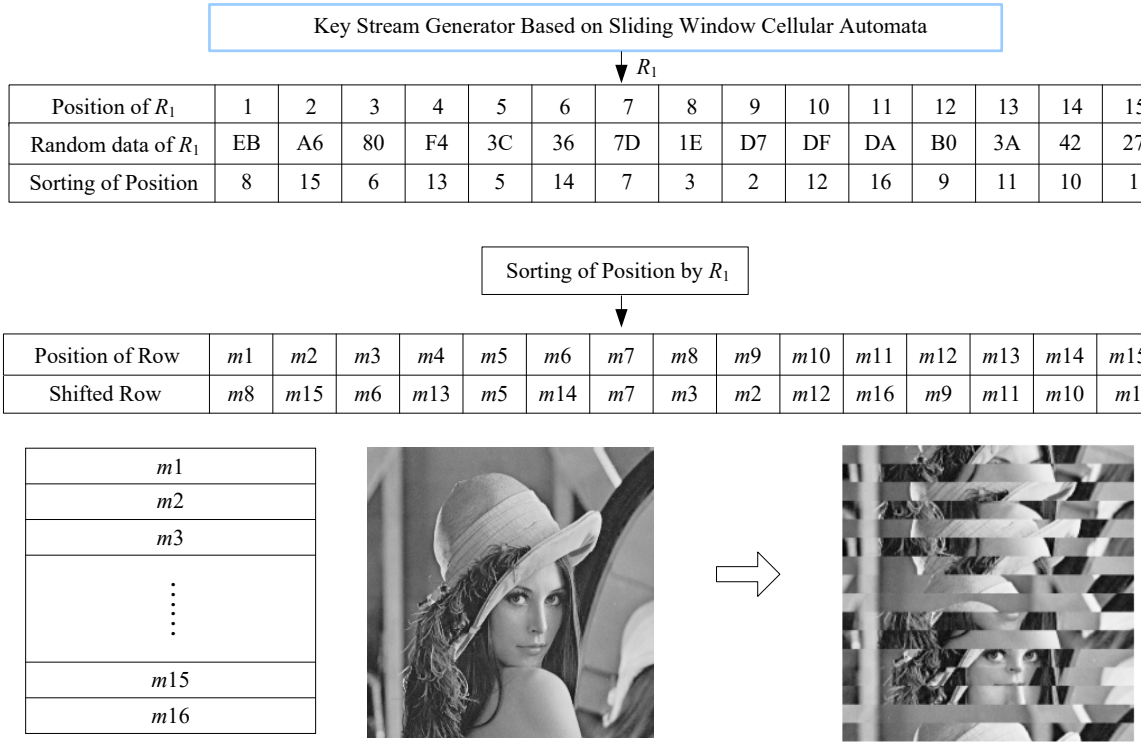
The main purpose of column scrambling and row scrambling is to disturb the pixels in each area of the image to change the overall structure of the image. Column and row scrambling can solve the problem of the same area of the image data having similar grayscale values and is characterized by high speed and efficiency. Row scrambling divides the image horizontally. We use the proposed SWCA-based keystream to scramble the columns on the image, as shown in Figure 8, where {$m1$, $m2$, $m3$, … $m16$} represents the data block after image segmentation. R_1 is a random keystream in bytes generated by the proposed keystream generator. The image value will be in rows arranged in ascending order through the keystream R_1. The image is divided vertically, and the generated keystream scrambles the columns, as shown in Figure 9, where {$n1$, $n2$, $n3$, … $n16$} represents the data block after image segmentation. R_2 is a random keystream in bytes, and the image segment will be arranged in ascending order through the keystream R_2 in column.

3.2.2 Confusion of Pixel Position

After the rows and columns are scrambled, the relative positions of pixels in the image blocks have changed but the image still retains some features. The confusion of pixel positions permutates the locations of pixels in the image, which can diffuse the position of the image pixels more finely to encrypt the image. The proposed confusion of pixel position is shown in Figure 10. We use the keystream produced by the proposed keystream generator to permutate the pixels, where {$p1$, $p2$, $p3$,… $p16$} represent the 16 pixels of the image, R_3 is a random keystream in units of bytes, and the image pixels are arranged in ascending order through the keystream R_3.

3.2.3 Confusion of Pixel Value

After the image is encrypted by scrambling rows and columns and confusing the pixel positions, for a flat image, only the image pixel positions are changed. Some traditional image encryption methods use scrambling, and only the pixels are moved. However, such images are easy to attack by statistical analysis because the histograms of the original image are the same as that of the scrambled image. To prevent statistical analysis attacks, it is necessary to confuse the pixel values [28]. In the following, we use the proposed keystream to confuse the pixel values on the image, as shown in Figure 11. First, the proposed keystream generator produces the keystream. Then, an exclusive OR operation is performed on the pixels with the keystream. Finally, the encrypted image is obtained, where R_4 is the keystream generated by the proposed keystream generator. P_i represents the pixels of the image, and C represents the encrypted image.



Position of R_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Random data of R_1	EB	A6	80	F4	3C	36	7D	1E	D7	DF	DA	B0	3A	42	27	C0
Sorting of Position	8	15	6	13	5	14	7	3	2	12	16	9	11	10	1	4

Position of Row	$m1$	$m2$	$m3$	$m4$	$m5$	$m6$	$m7$	$m8$	$m9$	$m10$	$m11$	$m12$	$m13$	$m14$	$m15$	$m16$
Shifted Row	$m8$	$m15$	$m6$	$m13$	$m5$	$m14$	$m7$	$m3$	$m2$	$m12$	$m16$	$m9$	$m11$	$m10$	$m1$	$m14$

Figure 8. The results of row scrambling

Key Stream Generator Based on Sliding Window Cellular Automata

R_2

Position of R_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Random data of R_2	EF	4E	82	B9	54	7A	9E	E7	52	9E	14	FD	31	31	FF	3B
Sorting of Position	11	13	16	14	2	9	5	6	3	7	10	4	15	8	1	12

Sorting of Position by R_2

Position of Column	$n1$	$n2$	$n3$	$n4$	$n5$	$n6$	$n7$	$n8$	$n9$	$n10$	$n11$	$n12$	$n13$	$n14$	$n15$	$n16$
Shifted Column	$n11$	$n13$	$n16$	$n14$	$n2$	$n9$	$n5$	$n6$	$n3$	$n7$	$n10$	$n4$	$n15$	$n8$	$n1$	$n12$

Figure 9. The results of column scrambling

Key Stream Generator Based on Sliding Window Cellular Automata

R_3

Position of R_3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Random data of R_3	2D	7A	C4	70	10	B9	66	7D	AA	DF	83	C	F9	79	D5	38
Sorting of Position R_3	12	5	1	16	7	4	14	2	8	9	11	6	3	15	10	13

Sorting of Position by R_3

Position of pixels	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p8$	$p9$	$p10$	$p11$	$p12$	$p13$	$p14$	$p15$	$p16$
Position of Shift Pixels	$p12$	$p5$	$p1$	$p16$	$p7$	$p4$	$p14$	$p2$	$p8$	$p9$	$p11$	$p6$	$p3$	$p15$	$p10$	$p13$

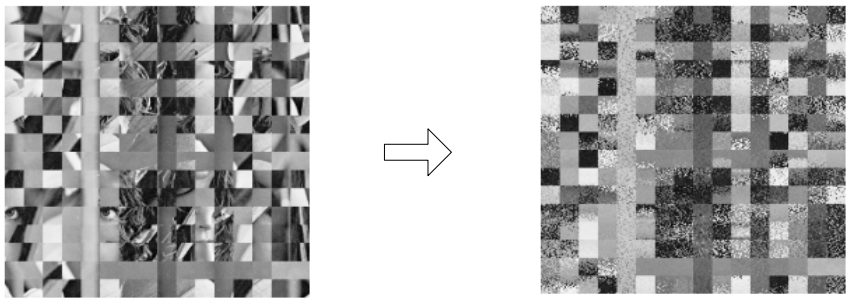


Figure 10. The result of confusion of pixel position

Key Stream Generator Based on Sliding Window Cellular Automata

R_4

Position of R_4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Random data of R_4	A9	39	6E	51	A5	FF	19	54	F7	85	46	C6	6D	38	60	71

P_i

Position of P	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P in hex	70	67	71	69	6A	6F	6C	D5	C9	70	69	69	6C	6E	66	69

$P \oplus R_4$

Position of C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C in hex	D9	5E	1F	38	CF	90	75	81	3E	F5	2E	AF	1	86	6	24
Data value of C	217	94	31	56	207	144	117	129	62	245	47	175	1	56	6	18

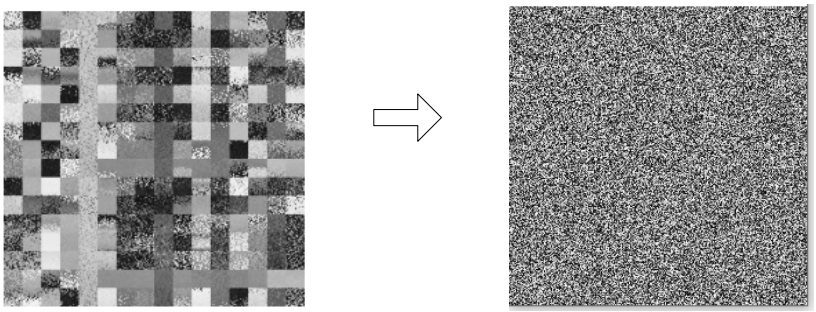
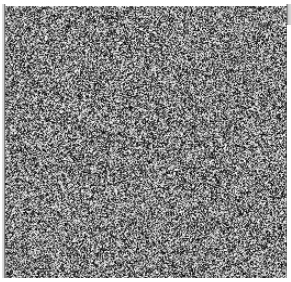


Figure 11. The result of the confusion of the pixel values

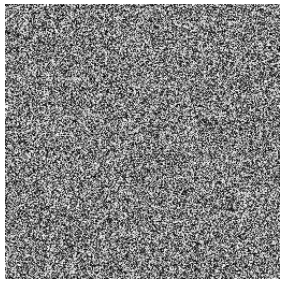
(a) Plaintext image of Lana



(b) Encrypted image of (a)

(c) Decrypted image of (a)

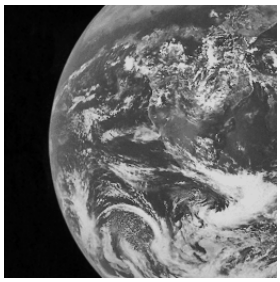
(d) Plaintext image of Ken



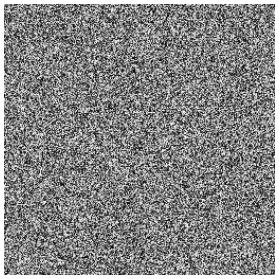
(e) Encrypted image of (d)

(f) Decrypted image of (d)

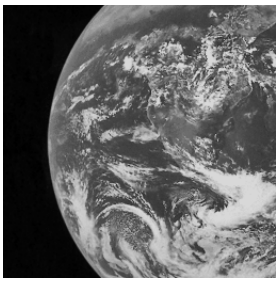
Figure 12. Experimental results of image encryption based on the proposed SWCA scheme



(g) Plaintext image of Earth

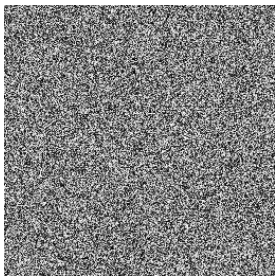


(h) Encrypted image of (g)



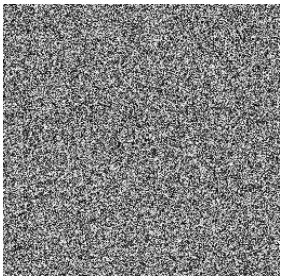
(i) Decrypted image of (g)

(j) Plaintext image of girl



(k) Encrypted image of (j)

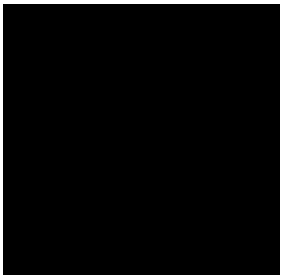
(l) Decrypted image of (j)



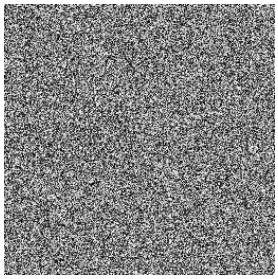
(m) Plaintext image of all-white

(n) Encrypted image of (m)

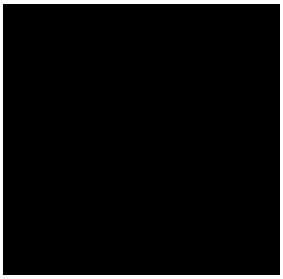
(o) Plaintext image of (m)



(p) Plaintext image of all-black



(q) Encrypted image of (p)



(r) Decrypted image of (p)

Figure 12. Experimental results of image encryption based on the proposed SWCA scheme (continued)

3.3 Experimental Results of the Proposed SWCA-based Image Encryption Scheme

In this section, we will show the experimental results of the image encryption system based on the proposed SWCA scheme. Our plaintext is a grayscale image of 256 × 256 pixels with a grayscale value of 0~255. After the proposed image encryption operation, i.e., by scrambling rows and columns and confusing the pixel positions and pixel values, the image ciphertext is obtained. Figure 12 is the results of plaintext image encryption, and image decryption.

4 Security Analysis and Experimental Results

In this section, we simulate the proposed SWCA-based image encryption system based on the proposed keystream generator. We also analyze the security of the simulated experimental results. The simulation environment of this experiment is shown in Table 2.

Table 2. Simulation environment

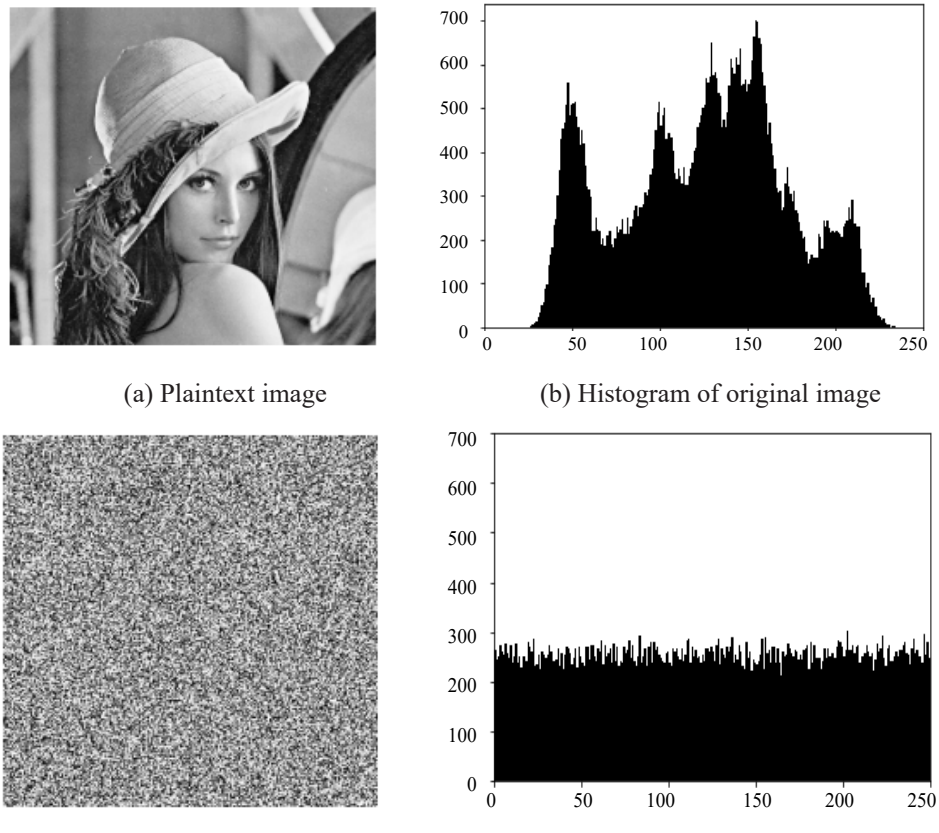
Operation system	Windows 10
CPU	AMD Ryzen 5 3400G 3.70 GHz
Memory	16.0 GB
Simulation system	Python 3.9

SP800-22 is a keystream test standard formulated by the National Institute of Standards (NIST) [14]. It is one of the most representative keystream tests. There are 15 tests in this statistical test. The keystream generated by the proposed scheme pass the tests.

Bit diffusion and image scrambling are not enough for secure image encryption. The pixel values in the same area will be so close that an attacker will be able to infer the original image by analyzing the distribution of image pixels. The reason is that the pixels of the original image and the encrypted image have the same distribution. In this section, we introduce the results of security analysis of the proposed image encryption system.

4.1 Histogram Analysis

Image histogram analysis is a method used to represent the distribution of pixels in a digital image. According to the total number of pixels in this image, a histogram of this image can be drawn. Through the histogram, we can easily understand the pixel distribution of the image. Firstly, we select a gray-scale image with a size of 256 × 256 pixels and compare the original image with the encrypted image by the proposed image encryption system. Both histogram analysis results are shown in Figure 13(b) and 13(d), where the horizontal axis of the histogram is the gray value of the image and the vertical axis is the number of pixels. And Figure 13(a) is the original image, and Figure 13(c) is the encrypted image.



(a) Plaintext image (b) Histogram of original image

(c) Encrypted image (d) Histogram of encrypted image

Figure 13. Results of histogram analysis

Figure 13 shows that the histogram of the original image is unevenly distributed, which means that the correlation of the original image is high. The histogram of the encrypted image is almost uniformly distributed. This means that, after the encryption, the position and value of the image have a good scrambling and confusion effect, which breaks the high correlation of the original image, making the image less vulnerable to statistical attacks by histogram analysis.

Table 3. Correlation coefficient between plaintext image and ciphertext image

Direction	Coefficients		
	Horizonal	Vertical	Diagonal
Plaintext image	0.92881	0.91888	0.92882
Encrypted image	0.00271	0.00244	0.00201

Table 4. Comparison of the correlation coefficients

Scheme	Coefficients		
	Horizontal	Vertical	Diagonal
Proposed scheme	0.00271	0.00244	0.00201
Chen's scheme [5]	0.0189	0.0201	0.0163
Zhang's scheme [19]	0.0086	0.00149	0.0046
Huang's scheme [29]	0.02119	0.00122	0.02953
Kamal's scheme [30]	-0.0093	0.0025	0.0042
Roy's scheme [31]	-0.005	-0.006	-0.003
Wang's scheme [32]	0.004778	-0.000248	0.000623
Zhang's scheme [33]	0.00201	0.00248	0.01323

4.2 Key Space Analysis

A good image encryption system needs a large enough key space to resist brute force attacks. The proposed image encryption system generates the keystreams R_1, R_2, R_3 and R_4 required for each stage by inputting the key *Key* to the proposed SWCA-based keystream generator to disrupt the entire image. The key *Key* is 128-bit binary data, and its key space is 2^{128}. The key space of the proposed image encryption system is large enough to resist the brute force attacks.

4.3 Correlation Analysis

The correlation coefficient can show the linear correlation between two variables x and y. The value of the coefficient is between 1 and -1. If x increases when y increases, the correlation coefficient is closer to 1, and if x decreases when y increases, the correlation coefficient is closer to -1. A correlation coefficient of 0 means that there is no linear correlation between the two variables x and y. Equations (9–13) show the algorithm of the correlation coefficient r_{xy}, where r_{xy} represents the covariance of two variables x and y, $cov(x,y)$, divided by the product of their standard deviations $D(x)$ and $D(y)$, and N is the total number of pixels.

$$\mathrm{cov}(x, y) = E(x - E(x))(y - E(y)) \ . \tag{9}$$

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} . \tag{10}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \ , E(y) = \frac{1}{N}\sum_{i=1}^{N} y_i \ . \tag{11}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}[x_i - E(x)]^2 . \tag{12}$$

$$D(y) = \frac{1}{N}\sum_{i=1}^{N}[y_i - E(y)]^2 . \tag{13}$$

We will calculate the correlation between two adjacent pixels in the encrypted image, which includes three types of correlations: between two vertical adjacent pixels, between two horizontal adjacent pixels, and between two diagonal adjacent pixels. First, we randomly select 65280 pairs of adjacent pixels. Then, we calculate r_{xy} according to the correlation coefficient formula. The correlation coefficients of the vertical, horizontal, and diagonal adjacent pixels of the plaintext image and the ciphertext image are shown in Table 3. According to Table 3, the correlation coefficient of the plaintext image is close to 1, which means that the adjacent pixels of the original image are highly correlated. The adjacent pixels of the encrypted image are close to 0, which means that the adjacent pixels of the encrypted image have low correlation. Table 4 is a comparison of the correlation coefficients of our proposed scheme and other schemes. This table indicates that the image encrypted via our proposed scheme has low correlation coefficient between the plaintext image and the ciphertext image.

4.4 Differential Attack

Differential attack is a method of encrypting the plaintext with a small difference, analyzing the change in the ciphertext, and then inferring the key or the structure of the encryption system. Therefore, a well-encrypted image

should be vastly different from the original image in order to avoid differential attack. The strength of the resistance to differential attack can be evaluated by two parameters: Number of Changing Pixel Rate (*NPCR*) and Unified Averaged Changed Intensity (*UACI*). The ideal value of *NPCR* is above 99%, and the ideal value of *UACI* is above 33% [17]. The calculations of *NPCR* and *UACI* are shown in Equation (14) and Equation (15), respectively, where C_1 and C_2 are the encrypted images and W and H are the length and width, respectively, of the encrypted images in pixels.

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \quad \text{where } D(i,j) = \begin{cases} 0, C_1(i,j) = C_2(i,j) \\ 1, C_1(i,j) \neq C_2(i,j) \end{cases}. \tag{14}$$

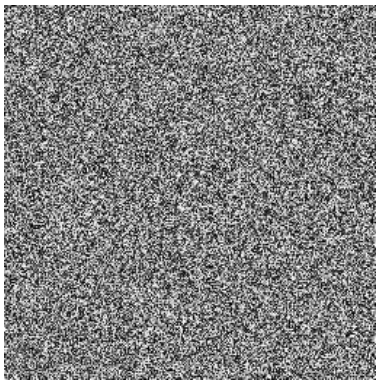
$$UACI \quad \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,j) \quad C_2(i,j)|}{255}\right] \times 100\% \ . \tag{15}$$

We perform *NPCR* and *UACI* calculation for the proposed image encryption system. In Figure 14, P_1 and P_2 are grayscale images of 256 × 256 pixels with a difference in pixels and C_1 and C_2 are the corresponding ciphertext images encrypted by P_1 and P_2. The test results of *NPCR* and *UACI* are shown in Table 5. For the proposed scheme, the *NPCR* value was 99.61% and the *UACI* value was 33.48%, which means that the proposed image encryption system can effectively resist a differential attack.

Table 5. Differential attack test results and comparisons

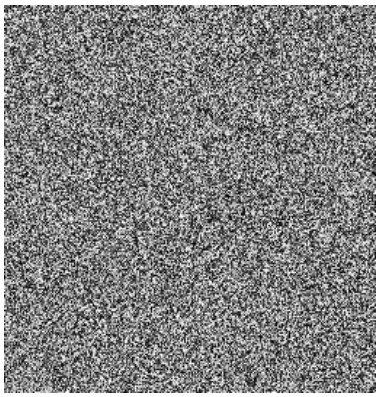
Scheme	NPCR	UACI
Proposed scheme	99.61%	33.48%
Zhang's scheme [19]	99.61%	33.49%
Kamal's scheme [30]	99.60%	33.43%
Roy's scheme [31]	99.93%	38.20%
Wang's scheme [32]	99.64%	33.41%
Zhang's scheme [33]	99.80%	33.18%

(a) Plaintext Image P_1



(b) The Encrypted Image C_1

(c) Plaintext Image P_2



(d) The Encrypted Image C_2

Figure 14. Test results of a differential attack

5 Conclusion

In this paper, we merge CA, sliding-window and bit permutation, and propose a sliding-window based CA keystream generator. We use the proposed keystream generated by SWCA for image encryption system. The image is scrambled and confused with the keystream, which effectively disrupts and covers the entire image. It has a larger key space. The histogram analysis and correlation analysis also show that the encrypted image has good pixel distribution and low correlation. In the differential attack analysis, the *NPCR* value and the *UACI* value are more than 99% and 33%, respectively. For the secure application of IoT, CA have the advantages of simplicity and modularization.

Acknowledgments

The author wishes to thank Guo-Hua Wu for his important assistance and the anonymous referees for their valuable and useful comments that have enriched this paper.

References

[1] S. Y. Moon, J. H. Park, J. H. Park, Authentications for Internet of Things Security: Threats, Challenges and Studies, *Journal of Internet Technology*, Vol. 19, No. 2, pp. 349-358, March, 2018.

[2] C.-C. Wu, R.-S. Cheng, C.-W. Hsu, L.-W. Wu, Lightweight, Low-Rate Denial-of-Service Attack Prevention and Control Program for IoT Devices, *Journal of Internet Technology*, Vol. 20, No. 3, pp. 877-885, May, 2019.

[3] S. Roy, N. Bhatia, U. S. Rawat, A Novel Cryptosystem Using Cellular Automata, *2017 International Conference on Communication and Signal Processing*, Chennai, India, 2017, pp. 1781-1785.

[4] S. Roy, U. Rawat, J. Karjee, A Lightweight Cellular Automata Based Encryption Technique for IoT Applications, *IEEE Access*, Vol. 7, pp. 39782-39793, March, 2019.

[5] Q. Chen, Y. Dai, Z. Niu, An Image Encryption Algorithm Based on Combination of Chaos and DNA Encoding, *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, Chongqing, China, 2020, pp. 182-185.

[6] E. Göncü, A. Koçdoğan, M. E. Yalçin, A High Speed True Random Number Generator with Cellular Automata with Random Memory, *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, Florence, Italy, 2018, pp. 1-5.

[7] HEM Infosec, Ltd, *OWASP Top 10 2021*, https://www.heminfosec.com/eDM/2021-10-OWASPtop10-2021-WAPPLES-OSCAN.html, 2021.

[8] A. Kumaravel, O. O. N. Meetei, An application of non-uniform cellular automata for efficient cryptography, *2013 IEEE Conference on Information & Communication Technologies*, Thuckalay, India, 2013, pp. 1200-1205.

[9] K. Rajeshwaran, K. A. Kumar, Cellular Automata Based Hashing Algorithm (CABHA) for Strong Cryptographic Hash Function, *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Coimbatore, India, 2019, pp. 1-6.

[10] K. J. J. Kumar, K. C. K. Reddy, S. Salivahanan, Novel and Efficient Cellular Automata Based Symmetric Key Encryption Algorithm for Wireless Sensor Networks, *International Journal of Computer Applications*, Vol. 13, No. 4, pp. 30-37, January, 2011.

[11] S. Nandi, B. K. Kar, P. P. Chaudhuri, Theory and Applications of Cellular Automata in Cryptography, *IEEE Transactions on Computers*, Vol. 43, No. 12, pp. 1346-1357, December, 1994.

[12] R. Dogaru, I. Dogaru, Efficient and Cryptographically Secure Pseudorandom Number Generators Based on Chains of Hybrid Cellular Automata Maps, *2014 10th International Conference on Communications*, Bucharest, Romania, 2014, pp. 1-4.

[13] Federal Information Processing Standards Publications, *Security Requirements for Cryptographic Modules*, FIPS PUB 140-1, January,1994.

[14] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, N. Heckert, J. Dray, S. Vo, L. Bassham, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, National Institute of Standards and Technology, NIST Special Publication 800-22 Revision 1, August, 2008

[15] H. Jiang, C. Fu, An Image Encryption Scheme Based on Lorenz Chaos System, *2008 Fourth* International *Conference on Natural Computation*, Jinan, China, 2008, pp. 600-604.

[16] J. Miano, *Compressed Image File Formats: Jpeg, Png, Gif, Xbm, Bmp*, Addison-Wesley Professional, 1999, pp. 23-30.

[17] Y. Wang, K. Wong, X. Liao, T. Xiang, G. Chen, A Chaos Based Image Encryption Algotithm with Variable Control Parameter, *Chaos, Solitons & Fractals*, Vol. 41, No. 4, pp. 1773-1783, August, 2009.

[18] R. Yin, J. Yuan, Q. Yang, X. Shan, X. Wang, Discretization of Coupled Map Lattices for a Stream Cipher, *Tsinghua Science & Technology*, Vol. 16, No. 3, pp. 241-246, June, 2011.

[19] Z. Zhang, S. Sun, Image encryption algorithm based on Logistic Chaotic System and s-box scrambling, *2011 4th International Congress on Image and Signal Processing*, Shanghai, China, 2011, pp. 177-181.

[20] H. Prasetyo, C.-H. Hsia, J.-Y. Deng, Multiple Secret Sharing with Simple Image Encryption, *Journal of Internet Technology*, Vol. 21, No. 2, pp. 323-341, March, 2020.

[21] S. L. Narayanan, K. Sankaranarayanan, V. Vijayakumari, A Secured and Accessibility Controlled Sharing of Images with Multiple Users, *Journal of Internet Technology*, Vol. 21, No. 2, pp. 469-478, March, 2020.

[22] Y.-C. Chen, C.-W. Shiu, Distributed Encrypted Image-Based Reversible Data Hiding, *Journal of Internet Technology*, Vol. 22, No. 1, pp. 101-107, January, 2021.

[23] J. V. Neumann, *Theory of Self-reproducing Automata*,

Edited by A. W. Burks, University of Illinois Press, 1966.
[24] S. Wolfram, Cryptography with Cellular Automata, in: H. C. Williams (Eds.), *Conference on the Theory and Application of* Cryptographic *Techniques*, Springer, Berlin, Heidelberg, 1986, pp. 429-432.
[25] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 3rd Edition, Prentice-Hall Inc, Upper Saddle River, N. J., 2003.
[26] S. Roy, M. Shrivastava, U. Rawat, C. Vinodkumar Pandey, S. K. Nayak, IESCA: An efficient image encryption scheme using 2-D cellular automata, *Journal of Information Security and Applications*, Vol. 61, Article No. 102919, September, 2021.
[27] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 5th Edition, Prentice Hall, 2008.
[28] P. Ping, J. Li, Y. Mao, R. Qi, Image Encryption Algorithm Based on Chaotic Maps and Bit Reconstruction, *Journal of Image and Graphics*, Vol. 22, No. 10, pp. 1348-1355, September, 2017.
[29] M. Huang, Y. Huang, Mingshi Wang, Image Encryption Algorithm Based on Chaotic Maps, *2010 International Computer Symposium (ICS2010)*, Tainan, Taiwan, 2010, pp. 154-158.
[30] S. T. Kamal, K. M. Hosny, T. M. Elgindy, M. M. Darwish, M. M. Fouda, A New Image Encryption Algorithm for Grey and Color Medical Images, *IEEE Access*, Vol. 9, pp. 37855-37865, March, 2021.
[31] M. Roy, S. Chakraborty, K. Mali, S. Mitra, I. Mondal, R. Dawn, D. Das, S. Chatterjee, A Dual Layer Image Encryption using Polymerase Chain Reaction Amplification and DNA Encryption, *2019 International Conference on Opto-Electronics and Applied Optics (Optronix)*, Kolkata, India, 2019, pp. 1-4.
[32] X. Wang, J. Yang, A Privacy Image Encryption Algorithm Based on Piecewise Coupled Map Lattice with Multi Dynamic Coupling Coefficient, *Information Sciences*, Vol. 569, pp.217-240, August, 2021.
[33] H. Zhang, X. Wang, H. Xie, C. Wang, X. Wang, An Efficient and Secure Image Encryption Algorithm Based on Non- Adjacent Coupled Maps, *IEEE Access*, Vol. 8, pp. 122104-122120, July, 2020.

Biography

Shyi-Tsong Wu was born in Jiaoxi, Yilan, Taiwan. He received his Ph.D. in the Department of Electronic Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2005. He is now an Associate Professor at the Department of Electronic Engineering, National Ilan University, Yilan City, Taiwan. His research interests include IoT security and applications, cryptography, and electronic circuits.