2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010

CA-based Stream Cryptography with Variable-Length Key

Masaki Igarashi, Masayuki Ikebe and Junichi Motohisa

Graduate School of Information Science and Technology, Hokaido University
14 Nishi, 9 Kita, Sapporo, 060-0814 Japan
Email: igarashi@impulse.ist.hokudai.ac.jp, ikebe, motohisa@ist.hokudai.ac.jp

Abstract—Cellular automata (CA) stream cryptography with a variable-length key was developed. CA are used to design secret key cryptography systems based on a one-time pad cipher. To change the connection of cells of one-dimensional CA which generate pseudo-random number sequences (PNS), the state of the cells can be shifted and the key length of the cryptography can also be controlled. Arranging combinations and mixing CA-chaos sequences in time and space conceals a feature of the CA-chaos sequences, and their frequency characteristics are similar to those of white noise. Moreover, pseudo-random-numbers are extracted with consideration of spatial direction and the random numbers are selected by the internal state of the cipher system for internal-state-recovery attacks. The cipher system was simulated, and the encryption/decryption throughput was 64 bit/cycle.

1. Introduction

In a chaos-stream cipher, a chaos generator combining a one-way function and is used as a random number generator [1]. Usually, a logistic map is used for the chaos generator and the chaos value is binalized as a one-way function. However, a symbolic dynamics attack[2], which analyzes a chaos generating mechanism from the long PNS, is proposed for this type of chaos cipher.

In Cellular Automata is used as random sequence generator[3]. Gutowitz uses Cellular Automata as discrete dynamical system to add complexity of the cryptosystem[4]. But none of these schemes has been able to withstand the modern attacks developed out of the cryptanalysis techniques. Rules of radius $r = 1$ and 2 for non-uniform 1-D CA have been also proposed for high quality random numbers generation[5]. However, peculiar patterns with the CA rules appear near the cell. Here, we proposed combining and mixing the CA chaos sequences to conceal the peculiar patterns, and developed a cipher system using one-dimensional CA with a variable-length key.

Figure 1 is a block diagram of a cellular automaton cipher system. It consists of a converter with a one-way function machine and a cellular automaton chaos generator corresponding to a variable-length key. Details are given below.

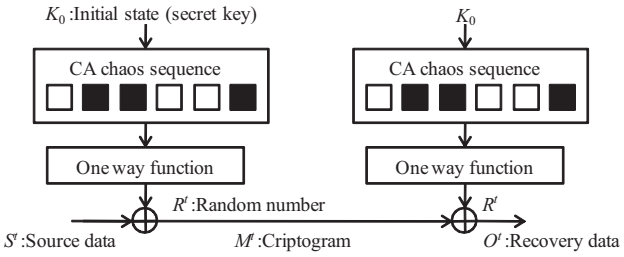

Figure 1: Block diagram of a CA cipher system.

2. Variable-Length Key of CA Cryptography

Here, we describe our method of creating the variable-length key by changing connections between cells. In conventional CA cell connections, an initial value is given to the one-dimensional cell array connected as a ring, and the cell array generates chaos by a appropriate rule. Here, we consider making the connection of the cells variable.

A bypass is added to the cell array with connection as a cross point, and it is made the shape of a small ring (Fig. 2). When the same initial values were given to the conventional cellular connection and the connection with a shifted cross point, different types of chaos were generated. In this way, a variable-length key can be made by shifting the knot of a bypass (the key length bit width is determined by the position the knot). Moreover, since the number of cells is not changed, the width of the obtained binary sequence does not change, and it becomes possible to conceal the composition of the chaos-generating mechanism from the outside.

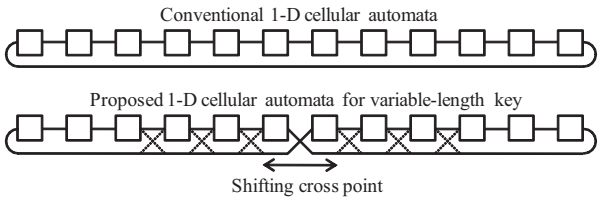

Figure 2: Implementation of variable-length key on the CA.

3. Proposed Random Number Generator with CA Chaos

A chaos-generating mechanism is described by temporal differentiation, which represents correlation of a time series. This is applied also to cellular automata. Correlation of a time series is also represented for cellular automata by the cell-interaction rule. Therefore, when the generated chaos value is used as it is, there is a weakness that the initial chaos-generating value will be found from a correlation of a time series. Particular patterns of 1-D CA also become the weakness for searching internal state. Here, a converter that took safety into consideration is realized using composition and a mixture of chaos and XORing.

The following three elements are applied to our random number generator.

- Two chaos sequences are combined.
- Mixing the chaos sequences on the time-space and compressing them.
- Generating two binary sequences from the compressed chaos and selecting one sequence at the time of generation of a cryptogram.

First, two one-dimensional cell arrays are prepared and each bit of two generated chaos sequences are compounded by XORing (Fig. 3 and Eqs. 1-2). Here, we used rule[30 and 86]. CA-state C_X^t consists of a set of each cell state $c_{X,k}^t$. t and k denote cell transition and cell position.

$$\begin{aligned} \boldsymbol{C_{30}^t} &= \{c_{30,0}^t, c_{30,1}^t, \cdots, c_{30,k}^t, \cdots, c_{30,127}^t\} \\ \boldsymbol{C_{86}^t} &= \{c_{86,0}^t, c_{86,1}^t, \cdots, c_{86,k}^t, \cdots, c_{86,127}^t\} \end{aligned} \tag{1}$$

$$\boldsymbol{R_0^t} = \boldsymbol{C_{30}^t} \oplus \boldsymbol{C_{86}^t} = \{c_{30,0}^t \oplus c_{86,0}^t, \cdots, c_{30,127}^t \oplus c_{86,127}^t\} \tag{2}$$

Next, a synthetic chaos sequence is arranged in a form suitable for mixing and compressing on a time axis (Fig. 4 and Eq. 3). For 128-bit width, the chaos sequence bends at 90 degrees and is generated with a length of 64 bits so that the length and the width become 64 bits. Then, the 128×64-bit sequence is separately overlapped with the two sequences of 64×64 bits. Each bit of these overlapping sequences is combined by XORing. By this conversion, a 128×64-bit series compresses to 64×64 bits.

$$\begin{aligned} \boldsymbol{R_1^t} &= \begin{pmatrix} r_{0,63}^t & \cdots & r_{0,63}^{t+63} \\ \vdots & & \vdots \\ r_{0,0}^t & \cdots & r_{0,0}^{t+63} \end{pmatrix} \oplus \begin{pmatrix} r_{0,64}^t & \cdots & r_{0,127}^t \\ \vdots & & \vdots \\ r_{0,64}^{t+63} & \cdots & r_{0,127}^{t+63} \end{pmatrix} \\ &= \begin{pmatrix} r_{1,0,0}^t & \cdots & r_{1,1,64}^t \\ \vdots & & \vdots \\ r_{1,64,0}^t & \cdots & r_{1,64,64}^t \end{pmatrix} \end{aligned} \tag{3}$$

Figure 5 and equations 4-5 show operation flow of the final stage processing. The generated 64×64-bit sequence

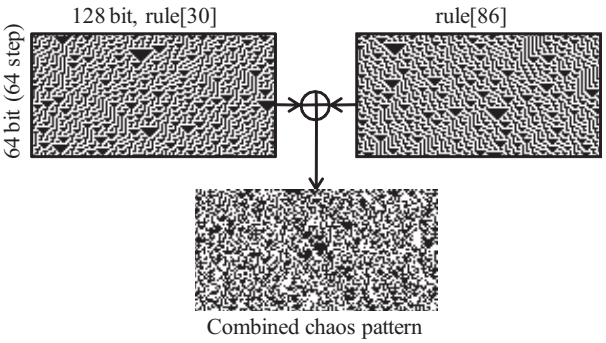

Figure 3: Combining of two chaos sequences

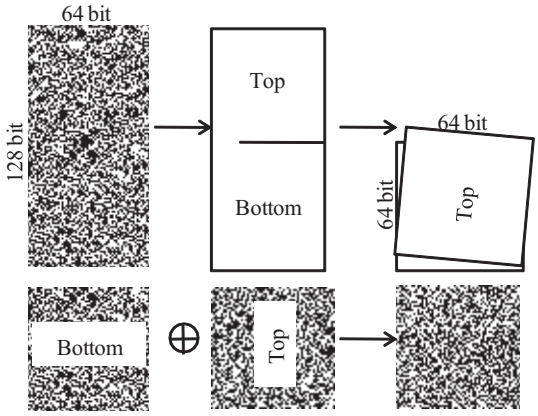

Figure 4: Mixing and compressing of chaos sequences

$\boldsymbol{R_1^t}$ is divided into two binary patterns. Two 64-bit patterns ($\boldsymbol{A^t}$ and $\boldsymbol{B^t}$) are extracted from each pattern by $Mask_A$ and $Mask_B$. Separately, from the 64x64-bit sequence $\boldsymbol{R_1^t}$, 64 bits on a diagonal line are extracted and a parity operation is performed on the bits. From the results of the parity operation $\boldsymbol{Parity^t}$, a 64-bit pattern is selected from $\boldsymbol{A^t}$ and $\boldsymbol{B^t}$ at the time of encryption. Since the 64 bits on the diagonal line and the result of the parity operation $\boldsymbol{Parity^t}$ are concealed from the outside, it cannot be confirmed from the outside whether $\boldsymbol{A^t}$ and $\boldsymbol{B^t}$ was selected.

$$\begin{aligned} \boldsymbol{A^t} &= \boldsymbol{R_1^t} \cdot Mask_A \\ \boldsymbol{B^t} &= \boldsymbol{R_1^t} \cdot Mask_B \end{aligned} \tag{4}$$

$$\boldsymbol{Parity^t} = \begin{cases} r_{1,0,0}^0 \oplus r_{1,1,1}^0 \oplus \cdots \oplus r_{1,63,63}^0 & (t=0) \\ O^{t-1} \oplus r_{1,0,0}^t \oplus r_{1,1,1}^t \oplus \cdots \oplus r_{1,63,63}^t & (t>0) \end{cases} \tag{5}$$

$$\begin{aligned} \boldsymbol{R^t} &= Sel(Parity^t, A^t, B^t) \\ Sel(Parity^t, A^t, B^t) &= \begin{cases} A^t(Parity^t = 1) \\ B^t(Parity^t = 0) \end{cases} \end{aligned} \tag{6}$$

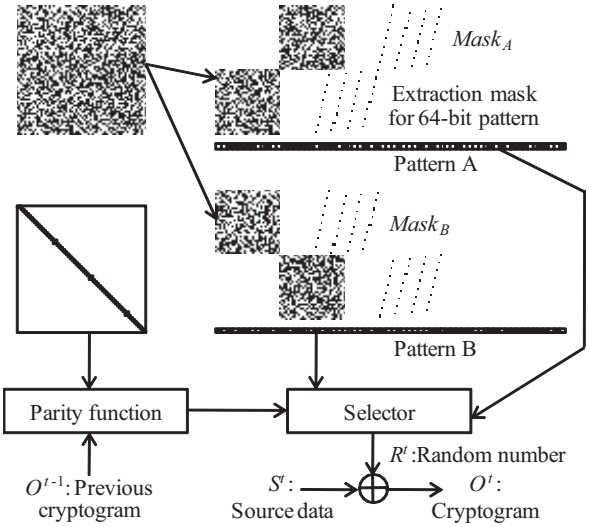

Figure 5: Operation flow of PNS generation

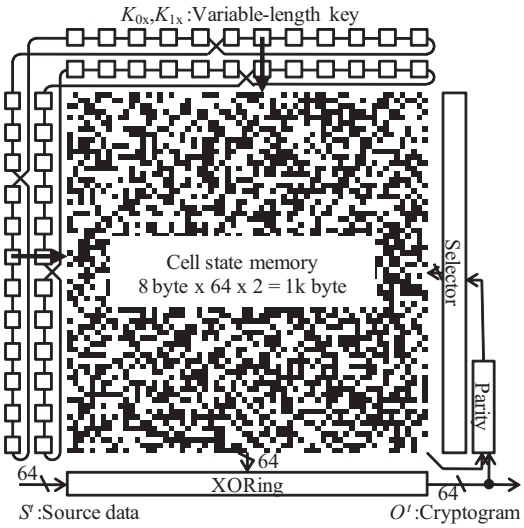

Figure 6: Proposed cipher system

4. Operation of Our Cipher System

Proposed cipher system is shown in Fig. 6. After the key setup is completed, the random number generation starts. Under processing, a chaos sequence is output for every cycle. After 128 operation cycles are completed, a 64-bit random number is generated for every cycle. First 64 operation cycles are used for a stable chaos generation. The 64-bit source data are enciphered at each cycle. After 192 operation cycles, the previous 64-bit-cryptogram unit is also used for selecting the 64-bit random number with the 64-bitparity operation on the diagonal line of a 64x64-bit-chaos pattern, and the system continues generating 64-bit random numbers. The random number generation of a proposal cipher system also takes into consideration the

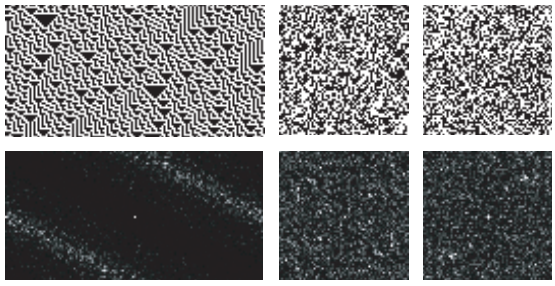

Figure 7: Fourier spectrum of chaos patterns (left: rule[86] (128×64 image), center: combined (bottom pattern, 64×64 image), right: mixed (64×64 image))

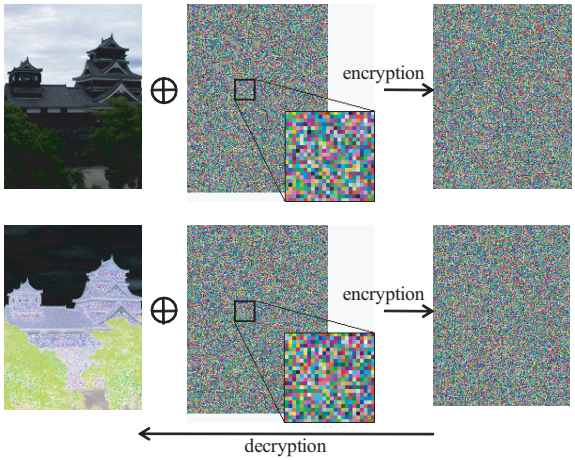

Figure 8: Simulation results of proposed cypher system

previous 64-bit cryptogram (i.e., that used for 64-bit selection). Therefore, if the data enciphered differs even if it uses the same key, a different random number will be generated. In decryption, the data is recovered in the same procedure using the same instrument and same key as encryption. Specification of our system is shown in Table 1.

5. Simulation Results

The results are shown in Figs. 7 and 8. Figure 7 show Fourier spectrum of chaos patterns. In the conventional CA sequence (rule [86]), particular patterns are confirmed and its frequency characteristic has also particular ones. How-

Table 1: Specification of proposed cipher system

Method	CA based chaos stream cipher
CA form	one dimensional, uniformed
rules	30, 86
Length of key	$32 \leq K_{0x}, K_{1x}, K_{0y}, K_{1y} \leq 64$
Throuthput	64 bit/cycle

ever, in the combined CA sequence, particular ones are reduced considerably and its frequency characteristic is close to white noise. Moreover, in the mixed CA sequence, its characteristics are improved.

Figure 8 shows results of our system operation. We confirmed that the encryption/decryption were performed normally and the difference patterns were generated with the initial images, even if the same key was used.

6. Consideration of Cipher Attack

In the proposed cipher system, two chaos sequences are combined. Here, the separation problem of chaos can be given to attackers. Moreover, in mixing and compressing of the chaos sequence, the chaos-generated values generated at different times are used. Since the data mixture on space time is applied to our system, this operation reduces the correlation of PNS[6]. In extracting the random number, a number with spatial distance is extracted and the internal state is concealed by selecting a number using an internal state.

There are four types of attacks that are assumed to be possible against the proposed system:

- Symbol dynamics attack
- Output prediction attack
- Internal state recovery attack
- Key recovery attack

The symbol dynamics attack searches for the sequence based on time series correlation of the binalized chaos value. In this type of attack, the approximate value of a higher bit is predicted in the state before binalization, and a brute force attack is applied to the lower bit. However, since a chaos of CA does not have distinct higher and lower bits, this attack cannot predict the approximate chaos value. Moreover, the attack cannot predict the sequence without time series correlation. Our system is protected from this type of attack by negating the time series correlation.

The remaining three types of attacks have relativity each other, and in our system, unless the internal state recovers, output prediction and key recovery cannot be performed. Therefore, recovery of the internal state passes through three processes:

- Recovery of selected PNS
- Decision of variable-length key positions
- Decision of an internal state

Each process is interlocking and each factor of the process is concealed. Each one requires calculation of 2^{64}, 2^{20} and 4^{4096} respectively. Therefore, if there is an internal state recovery attack against our system, computational complexity will increase more than it would with a brute force attack.

7. Conclusion

We proposed a method for CA-based stream cryptography with variable-length key and evaluated the cipher system. To change the connection of cells of one-dimensional CA, we controlled the key length of the cryptography. Arranging combinations and mixing CA-chaos sequences in time and space, we concealed a feature of the CA-chaos sequences, and we improved their frequency characteristics, which became similar to those of white noise. Moreover, using selection of PNSs are applied to our system, we protected the internal state of the system from the attacks. The cipher system was simulated, we conformed that the encryption/decryption throughput was 64 bit/cycle.

References

[1] Tsueike M, Ueta T, Nishio Y. ”An application of two-dimensional chaos cryptosystem.” Tech. Rep. of IEICE, NLP96-19, May 1996.

[2] K. Ohkuma and K. Sakurai, ”An Improvement of a Symbolic Dynamics Attack against a Chaos Stream Cipher [in Japanese]”, Proceedings of the IEICE General Conference, (1999), pp. 229.

[3] S. Wolfram, ”Cryptography with cellular automata, in: Advances in Cryptology” Crypto ’85 Proceedings, LNCS 218, Springer, (1986), pp. 429-432.

[4] H. Gutowitz, ”Cryptography with dynamical systems”, in: E. Goles, N. Boccara (Eds.), Cellular Automata and Cooperative Phenomena, Kluwer, 1993.

[5] F. Seredynski, P. Bouvry, A. Y. Zomaya, ”Cellular automata computations and secret key cryptography”, Parallel Computing, 30, (2004), pp. 753-766(2004).

[6] R. A. Rueppel, ”Correlation Immunity and the Summation Generator”, Crypto ’94 Proceedings, LNCS 839, Springer, (1994), pp. 411-424.