World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:3, No:12, 2009

# Linear cryptanalysis for a chaos-based stream cipher

Ruming Yin, Jian Yuan, Qiuhua Yang, Xiuming Shan, Xiqin Wang



*Abstract*—Linear cryptanalysis methods are rarely used to improve the security of chaotic stream ciphers. In this paper, we apply linear cryptanalysis to a chaotic stream cipher which was designed by strictly using the basic design criterion of cryptosystem – confusion and diffusion. We show that this well-designed chaos-based stream cipher is still insecure against distinguishing attack. This distinguishing attack promotes the further improvement of the cipher.

*Keywords*—Stream cipher, chaos, linear cryptanalysis, distinguishing attack.

## I. Introduction

In recent years, chaos has been used to design ciphers. Chaotic systems are defined on real numbers and characterized by sensitive dependence on initial conditions and parameters, random-like behavior, which are desirable to ciphers. The early chaotic ciphers were designed by directly using the chaotic maps. These ciphers often suffer from security weakness. The well-known cryptographic techniques were not effectively used to improve the security of the ciphers. On one hand, while the typical cryptographic operations are performed on binary numbers, the derived chaotic ciphers often consist of complicated operations on real numbers, thus the well-known cryptographic techniques are difficult to use. On the other hand, the existed well-known attacks such as differential and linear cryptanalysis were rarely considered in the design of chaotic ciphers to improve the security.

For enhanced security, *Kocarev* first presented that the chaotic ciphers should be designed by using strict cryptographic techniques [1]. Then, several chaotic block ciphers were designed to resist differential and linear cryptanalysis [2, 3]. In [4, 5], chaotic substitution boxes (S-boxes) were investigated by using differential and linear cryptanalysis. Recently, *Masuda et al.* used the structures of modern block cipher to design chaotic block ciphers [6]. However, since all these efforts focused on the design of chaotic block ciphers, the developed methods can not be easily used when designing chaotic stream ciphers.

In this paper, linear cryptanalysis is applied to a chaos-based stream cipher, which was designed by strictly using the basic design criterion of cryptosystem – confusion and diffusion and passed main security evaluation criterion [7]. The linear cryptanalysis leads to a distinguishing attack on the cipher. We find that the key stream generated by the cipher can be easily distinguished from a truly random sequence, which means that the stream cipher is insecure in the strict sense of security [8]. This distinguishing attack promotes the further improvement of the cipher. In this way, our work highlights the importance of strict cryptographic techniques in the design and cryptanalysis of chaotic stream cipher and illustrates the approaches to use them.

The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.
E-mail: yrm05@mails.tsinghua.edu.cn, jyuan@tsinghua.edu.cn.

## II. Description of the Cipher

The cipher that will be analyzed in this paper is based on coupled map lattice (CML) consisting of skew tent maps [7]. Since CML is discretized to operate on binary numbers by explicitly using the structure for confusion and diffusion, the security of this cipher can be easily evaluated by using proper cryptographic techniques. The key streams of the cipher pass various randomness tests. The period length of key stream is very large (more than $2^{64}$). Additionally, the cipher was proved to be secure against guess-and-determine attacks. Thus, the cipher has some desirable properties. We briefly describe the stream cipher below.

The cipher is a synchronous stream cipher which uses a 128-bit key. The internal state of the cipher contains a 128-bit state variable $X_n$ and a 128-bit counter $C_n$ at time $n$. $X_n$ and $C_n$ are divided into eight 16-bit substrings respectively: $X_n = x_{7,n}\|x_{6,n}\|\cdots\|x_{0,n}$ and $C_n = c_{7,n}\|c_{6,n}\|\cdots\|c_{0,n}$, where $\|$ denotes concatenation of two bit sequences. The cipher works in two phases: first, the state variable $X$ and the counter $C$ are initialized using the key by the key setup algorithm, then the state-update function is repeatedly iterated and the key streams are generated. Since our distinguishing attack is just based on the analysis of the state-update function and the key stream generation process, we just describe these parts in this paper.

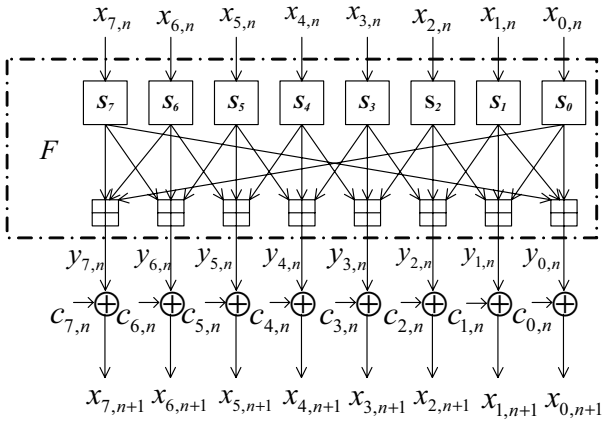


Fig. 1. The state-update function.

### A. The state-update function

The state-update function is shown in Fig. 1. It is designed as follows

$$X_{n+1} = F(X_n) \oplus C_n \tag{1}$$

where $''\oplus''$ denotes bitwise XOR. For the substrings of the internal state $X_n$ and $C_n$, the state-update function can be written as

$$\begin{aligned}
x_{0,n+1}&=y_{0,n} \oplus c_{0,n}\\
&=(S_7(x_{7,n}) \boxplus S_0(x_{0,n}) \boxplus S_1(x_{1,n})) \oplus c_{0,n},\\
x_{7,n+1}&=y_{7,n} \oplus c_{7,n}\\
&=(S_6(x_{6,n}) \boxplus S_7(x_{7,n}) \boxplus S_0(x_{0,n})) \oplus c_{7,n},\\
x_{j,n+1}&=y_{j,n} \oplus c_{j,n}\\
&=(S_{j-1}(x_{j-1,n}) \boxplus S_j(x_{j,n}) \boxplus S_{j+1}(x_{j+1,n})) \oplus c_{j,n},\\
&\quad j = 1, 2, \cdots, 6
\end{aligned} \tag{2}$$

where $''\boxplus''$ denotes addition modulo $M = 2^{16}$. $x_{j,n}$ and $c_{j,n}$ denote the $j$-th state variable substring and the counter substring at time $n$ respectively. $y_{j,n}$ is the output of the function $F$. $S_j : [0, M-1] \rightarrow [0, M-1]$ is a substitution function, which is obtained by iterating a function $T_j$ six times as follows

$$S_j(t) = T_j^6(t+1) - 1,\ t \in [0, M-1]. \tag{3}$$

$T_j$ is the discretized skew tent map defined as

$$T_j(\tau) = \begin{cases} \lceil \frac{M\tau}{A_j} \rceil & 1 \le \tau \le A_j \\ \lfloor \frac{M(M-\tau)}{M-A_j} \rfloor + 1 & A_j < \tau \le M \end{cases} \tag{4}$$

where the parameters are $A_0 = 16409$, $A_{j+1} = A_j - 2$, $j = 0, 1, \cdots, 6$.

In the state-update function, a counter is used to expand the cycle length of the key streams. Its value is updated at each iteration as follows

$$C = C + 1 \ mod\ 2^{128}. \tag{5}$$

### B. Key stream generation and encryption/decryption

After each iteration of the state-update function, 64-bit key streams $s_n = s_{3,n} \| s_{2,n} \| s_{1,n} \| s_{0,n}$ are generated as follows

$$s_{j,n} = x_{j+4,n} \oplus x_{j,n},\ j = 0, 1, 2, 3. \tag{6}$$

In the encryption phase, the plaintext is transformed into the ciphertext by the bitwise XOR of the key stream $s_n$ and the plaintext. The decryption is accomplished by applying the enciphering a second time.

## III. Linear correlations between consecutive key streams

In this section, we apply linear cryptanalysis to the stream cipher. Here, the linearity refers to bitwise XOR denoted by "$\oplus$". In our linear cryptanalysis, we try to find the linear correlations between bits in $s_n$ and $s_{n+1}$ expressed as

$$s_n^{[i_1]} \oplus s_n^{[i_2]} \oplus \cdots \oplus s_n^{[i_u]} \oplus s_{n+1}^{[j_1]} \oplus s_{n+1}^{[j_2]} \oplus \cdots \oplus s_{n+1}^{[j_v]} = 0. \tag{7}$$

where $s_n^{[i_u]}$ represents the $i_u$-th bit of $s_n$. When numbering bits of variables, the least significant bit is denoted by 0 throughout this paper. If the equation of the form (7) is found, then the distinguisher can be constructed by using the concept of linear cryptanalysis [9, 10].

Consider that if we randomly select values for $u+v$ bits and place them into equation (7), the probability that the expression holds, denoted by $p$, will be exactly 1/2. However, for the key streams generated by the cipher, $p$ will have a value different from 1/2. The bias from 1/2 of $p$, denoted by $\varepsilon = |p - 1/2|$, determines the effectiveness of the linear approximations. The higher the magnitude of $\varepsilon$, the fewer key streams, i.e., the fewer number of iterations of the cipher are required for the constructed distinguisher [9].

To find equation of the form (7), we first analyze the linear approximations of S-boxes and modular addition respectively. Then we concatenate these approximations to obtain the linear approximations of the state-update function. To simplify the notation, we will denote $s_n^{[i_1]} \oplus s_n^{[i_2]} \oplus \cdots \oplus s_n^{[i_u]}$ by $s_n^{[i_1, i_2, \cdots, i_u]}$ in the following description.

### A. Linear approximation of S-boxes

For the S-box $S : [0, 2^l - 1] \rightarrow [0, 2^l - 1]$, all the linear approximations between the input bits and output bits can be expressed as follows

$$\bigoplus_{h=0}^{l-1}(x^{[h]} \bullet \theta^{[h]})) = (\bigoplus_{g=0}^{l-1} S(x)^{[g]} \bullet \eta^{[g]})) \tag{8}$$

where $0 \le \theta \le 2^l - 1$, $0 \le \eta \le 2^l - 1$. $\bullet$ denotes a bitwise AND operation and $x^{[h]}$ is the $h$-th bit of $x$. In this paper, we just consider linear equations (8) containing single output bit, since such linear equations will make the constructed distinguisher involve fewer counter bits, thus make the distinguisher easy to apply (see section 4). That is to say, we select $\eta = 2^i$ $(i = 0, 1, \cdots, l-1)$ in (8), where $i$ is the output bit position. In this case, linear equation (8) can be rewritten as

$$S(x)^{[i]} = \bigoplus_{h=0}^{l-1}(x^{[h]} \bullet \theta^{[h]}). \tag{9}$$

We can compute the bias of the linear approximations of S-boxes with the method proposed in [10]. For each $i$, we compute the bias of linear equations for all possible values of $\theta$, then we get the largest $\varepsilon$ for the selected $i$. In Table 1, we show the largest bias, denoted by $\varepsilon_{S_0,i}$, against the output bit position $i$ for the S-box $S_0$. The results are very similar for the other S-boxes in the cipher. We will use these results to estimate the bias of linear approximations between consecutive key streams.

TABLE I
Maximum value of the probability bias of $S_0$.

| $i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $\varepsilon_{S_0,i}$ | 0.0223 | 0.0236 | 0.0424 | 0.0160 | 0.0340 | 0.0397 | 0.0427 | 0.0353 |
| $i$ | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| $\varepsilon_{S_0,i}$ | 0.0401 | 0.0482 | 0.0500 | 0.0544 | 0.0683 | 0.1195 | 0.1289 | 0.1190 |

### B. Linear approximation of modular addition

The modular addition used in the cipher is

$$f(a,b,c) = a + b + c \ mod \ 2^{16} \tag{10}$$

where $a$, $b$, $c$ are integers in the interval $[0, 2^{16}-1]$. According to [11], for each output bit position $i \geq 2$, one of the best linear approximations is

$$f^{[i]} = a^{[i]} \oplus b^{[i]} \oplus c^{[i]}, i = 0, 1, \cdots, 15. \tag{11}$$

The probability bias of these linear approximations are

$$\varepsilon_{m,i} = \begin{cases} 1/2 & i = 0 \\ 0 & i = 1 \\ \sum_{v=1}^{i-1} 2^{-2v-1} & 2 \leq i \leq 15 \end{cases}. \tag{12}$$

### C. Linear correlations between $s_n$ and $s_{n+1}$

We concatenate the linear approximations of S-boxes and modular addition to construct the equation of the form (7). According to (6), the key streams at time $n+1$ are

$$\begin{aligned} s_{j,n+1} &= x_{j+4,n+1} \oplus x_{j,n+1} \\ &= y_{j+4,n} \oplus c_{j+4,n} \oplus y_{j,n} \oplus c_{j,n} \\ &= y_{j+4,n} \oplus y_{j,n} \oplus c_{j+4,n} \oplus c_{j,n} \\ &= o_{j,n} \oplus c_{j+4,n} \oplus c_{j,n}, \\ & \quad j = 0, 1, 2, 3. \end{aligned} \tag{13}$$

In equation (13), we denote $y_{j+4,n} \oplus y_{j,n}$ by $o_{j,n}$ for compactness. Further, we define $o_n = o_{3,n} \| o_{2,n} \| o_{1,n} \| o_{0,n}$. We will create the effective linear approximations between $s_n$ and $s_{n+1}$ in two phase: first, we construct the linear approximations between $o_n$ and $s_n$, then the counter is taken into account and linear equations of the form (7) are created.

*1) Linear correlations between $o_n$ and $s_n$:* We start by considering bits in $o_{0,n}$. Following the state-update function (see Fig. 1), the $i$-th bit in $o_{0,n}$ can be written as

$$\begin{aligned} o_{0,n}^{[i]} &= y_{4,n}^{[i]} \oplus y_{0,n}^{[i]} \\ &= (S_{5,n} \boxplus S_{4,n} \boxplus S_{3,n})^{[i]} \oplus (S_{1,n} \boxplus S_{0,n} \boxplus S_{7,n})^{[i]} \end{aligned} \tag{14}$$

where $S_{j,n}$ denotes $S_j(x_{j,n})$, which is the output of the S-box $S_j$. By using the linear approximation of modular addition (equation (11)) two times, we obtain the following equation:

$$o_{0,n}^{[i]} = S_{5,n}^{[i]} \oplus S_{4,n}^{[i]} \oplus S_{3,n}^{[i]} \oplus S_{1,n}^{[i]} \oplus S_{0,n}^{[i]} \oplus S_{7,n}^{[i]}. \tag{15}$$

Then, the linear approximations of the S-boxes (equation (9)) are applied six times and the following linear approximations between bits in $o_{0,n}$ and $s_n$ are obtained:

$$\begin{aligned} o_{0,n}^{[i]} &= x_{5,n}^{[u_1,u_2,\cdots,u_{n_1}]} \oplus x_{4,n}^{[v_1,v_2,\cdots,v_{n_2}]} \oplus x_{3,n}^{[w_1,w_2,\cdots,w_{n_3}]} \\ &\quad \oplus x_{1,n}^{[u_1,u_2,\cdots,u_{n_1}]} \oplus x_{0,n}^{[v_1,v_2,\cdots,v_{n_2}]} \oplus x_{7,n}^{[w_1,w_2,\cdots,w_{n_3}]} \\ &= s_{0,n}^{[v_1,v_2,\cdots,v_{n_2}]} \oplus s_{1,n}^{[u_1,u_2,\cdots,u_{n_1}]} \oplus s_{3,n}^{[w_1,w_2,\cdots,w_{n_3}]}. \end{aligned} \tag{16}$$

Note that, as is demonstrated in equation (16), the linear approximations of the S-boxes $S_j$ and $S_{j+4}$ should involve the same input bits. In this case, $o_{0,n}^{[i]}$ can be approximated by linear equations just containing bits of $s_n$.

Now, we have constructed linear approximations between bits in $o_{0,n}$ and $s_n$. Similarly, linear approximations between bits in $o_{j,n}$ $(j = 1, 2, 3)$ and $s_n$ can be found. They are written as follows. For compactness, the linear approximations of S-boxes are not written out. For all these equations, the approximations of S-boxes are used six times and the approximations of modular addition are used two times.

$$o_{1,n}^{[i]} = S_{6,n}^{[i]} \oplus S_{5,n}^{[i]} \oplus S_{4,n}^{[i]} \oplus S_{2,n}^{[i]} \oplus S_{1,n}^{[i]} \oplus S_{0,n}^{[i]}. \tag{17}$$

$$o_{2,n}^{[i]} = S_{7,n}^{[i]} \oplus S_{6,n}^{[i]} \oplus S_{5,n}^{[i]} \oplus S_{3,n}^{[i]} \oplus S_{2,n}^{[i]} \oplus S_{1,n}^{[i]}. \tag{18}$$

$$o_{3,n}^{[i]} = S_{0,n}^{[i]} \oplus S_{7,n}^{[i]} \oplus S_{6,n}^{[i]} \oplus S_{4,n}^{[i]} \oplus S_{3,n}^{[i]} \oplus S_{2,n}^{[i]}. \tag{19}$$

In fact, other linear approximations between $o_n$ and $s_n$ can be formed by linear combination of the equations obtained. When combining the equations, the output bit of the S-boxes will be canceled out. Therefore, linear approximations of the S-boxes will be used fewer times. On the other hand, linear approximations of modular addition will be used more times. For example, by linear combination of equation (15) and (17), we get the following linear equation.

$$o_{1,n}^{[i]} \oplus o_{0,n}^{[i]} = S_{6,n}^{[i]} \oplus S_{3,n}^{[i]} \oplus S_{2,n}^{[i]} \oplus S_{7,n}^{[i]}. \tag{20}$$

Linear combination of arbitrary two equations of (15), (17), (18), (19) leads to a similar linear approximation. For all such kind of linear approximations, the approximations of S-boxes are used four times and the approximations of modular addition are used four times.

Further, linear approximations between $o_n$ and $s_n$ can be gotten by linear combination of arbitrary three equations of (15), (17), (18), (19). As an example, linear approximation by combining (15), (18), (19) is described as follows

$$o_{0,n}^{[i]} \oplus o_{2,n}^{[i]} \oplus o_{3,n}^{[i]} = S_{3,n}^{[i]} \oplus S_{7,n}^{[i]}. \tag{21}$$

In these linear approximations, the approximations of S-boxes are used two times and the approximations of modular addition are used six times.

We denote the times that linear approximations of modular addition and the S-boxes are used by $L_m$ and $L_S$ respectively. Then the linear approximations between bits in $o_n$ and $s_n$ can be sorted by $(L_m, L_S)$: $(L_m, L_S)$ = (2,6), (4,4) and (6,2). We will compute the probability bias of these three kinds of linear approximations and then select the largest one. According to Piling-Up Principle [10], the probability bias of the linear approximations between bits in $o_n$ and $s_n$ can be estimated as

$$\varepsilon_i = 2^7 (\varepsilon_{m,i})^{L_m} (\varepsilon_{S,i})^{L_S} \tag{22}$$

where $i$ $(i = 0, 1, \cdots, 15)$ represents the position of the output bit involved in the linear approximations. $\varepsilon_{m,i}$ and $\varepsilon_{S,i}$ are the probability bias of approximations of modular addition and the S-boxes respectively. Since the S-boxes in the cipher behave similarly, we just use the largest probability bias of S-box $S_0$ as an estimation of the probability bias of other S-boxes. That is to say, in our computation, $\varepsilon_{S,i} = \varepsilon_{S_0,i}$. By using (22), we compute $\varepsilon_i$ for all the three kinds of linear approximations between bits in $o_n$ and $s_n$ and for all different $i$. Then we get the best linear approximations with the largest bias. The results show that $\varepsilon_i$ gets the largest value, about

$2^{-10}$, when $(L_m, L_S) = (6, 2)$ and $i = 0$. Thus we obtain the best approximations between bits in $o_n$ and $s_n$, which are expressed as follows.

$$o_{0,n}^{[0]} \oplus o_{2,n}^{[0]} \oplus o_{3,n}^{[0]} = S_{3,n}^{[0]} \oplus S_{7,n}^{[0]}. \quad (23)$$
$$o_{0,n}^{[0]} \oplus o_{1,n}^{[0]} \oplus o_{2,n}^{[0]} = S_{1,n}^{[0]} \oplus S_{5,n}^{[0]}. \quad (24)$$
$$o_{0,n}^{[0]} \oplus o_{1,n}^{[0]} \oplus o_{3,n}^{[0]} = S_{0,n}^{[0]} \oplus S_{4,n}^{[0]}. \quad (25)$$
$$o_{1,n}^{[0]} \oplus o_{2,n}^{[0]} \oplus o_{3,n}^{[0]} = S_{2,n}^{[0]} \oplus S_{6,n}^{[0]}. \quad (26)$$

The probability bias of these best linear approximations is estimated to be about $2^{-10}$. In fact, since we require that the linear approximations of the S-boxes $S_j$ and $S_{j+4}$ involve the same input bits (see equation (16)), the actual value of the probability bias will be smaller. For example, by further analysis and computation of equation (23), the following linear approximations between bits in $o_n$ and $s_n$ is found. The probability that this equation holds is about 0.5004575, i.e., the probability bias is about $\varepsilon^{max} = 0.0004575$ $(\approx 2^{-11})$.

$$\begin{aligned} o_{0,n}^{[0]} \oplus o_{2,n}^{[0]} \oplus o_{3,n}^{[0]} &= S_{3,n}^{[0]} \oplus S_{7,n}^{[0]} \\ &= x_{7,n}^{[8,7,6]} \oplus x_{3,n}^{[8,7,6]} \oplus 1 \\ &= s_{3,n}^{[8,7,6]} \oplus 1. \end{aligned} \quad (27)$$

Since this linear approximation make the distinguisher much easier to apply when the counter is taken into account, it will be used to construct the distinguisher.

*2) Linear correlations between $s_n$ and $s_{n+1}$:* In this part, we take the counter into consideration and construct the linear correlations between consecutive key streams $s_n$ and $s_{n+1}$.

By using (13), equation (23), (24), (25), (26) can be written as

$$s_{0,n+1}^{[0]} \oplus s_{2,n+1}^{[0]} \oplus s_{3,n+1}^{[0]} = S_{3,n}^{[0]} \oplus S_{7,n}^{[0]} \oplus c_{[4,0,6,2,7,3],n}^{[0]} \quad (28)$$
$$s_{0,n+1}^{[0]} \oplus s_{1,n+1}^{[0]} \oplus s_{2,n+1}^{[0]} = S_{1,n}^{[0]} \oplus S_{5,n}^{[0]} \oplus c_{[4,0,5,1,6,2],n}^{[0]} \quad (29)$$
$$s_{0,n+1}^{[0]} \oplus s_{1,n+1}^{[0]} \oplus s_{3,n+1}^{[0]} = S_{0,n}^{[0]} \oplus S_{4,n}^{[0]} \oplus c_{[4,0,5,1,7,3],n}^{[0]} \quad (30)$$
$$s_{1,n+1}^{[0]} \oplus s_{2,n+1}^{[0]} \oplus s_{3,n+1}^{[0]} = S_{2,n}^{[0]} \oplus S_{6,n}^{[0]} \oplus C_{[5,1,6,2,7,3],n}^{[0]} \quad (31)$$

Where $c_{[r_1,r_2,\cdots,r_6],n}^{[0]}$ represents $c_{r_1,n}^{[0]} \oplus c_{r_2,n}^{[0]} \oplus \cdots \oplus c_{r_6,n}^{[0]}$ for compactness. It is noted that the number 1 is added modulo $2^{128}$ to the counter value after each iteration of the state-update function. Therefore, to make the above linear approximations hold with the expected probability, we must predict the counter value correctly. In other words, we should select the linear approximations with the counter bits easily predicted.

The probability bias of these linear approximations is about $\varepsilon^{max} = 2^{-11}$. Following the results in [9], the number of iterations of the cipher needed for the distinguisher is some small multiple of $1/(\varepsilon^{max})^2 \approx 2^{22}$ (see also section 4). In this case, we find that the counter bits involved in equation (28) are easily predicted. According to (7), the least significant bit $c_{0,n}^{[0]}$ in equation (28) is complemented after each iteration. The other counter bits keep unchanged with a probability of about $1 - 2^{22}/2^{32} = 0.99$ in consecutive $2^{22}$ iterations of the cipher, i.e., they are nearly fixed for our distinguisher. Thus, we can predict all the counter bits just by guessing a single bit. By using (27), equation (28) can be written as

$$s_{0,n+1}^{[0]} \oplus s_{2,n+1}^{[0]} \oplus s_{3,n+1}^{[0]} = s_{3,n}^{[8,7,6]} \oplus c_{[4,0,6,2,7,3],n}^{[0]} \oplus 1. \quad (32)$$

This equation can be further written in the form of (7) as

$$s_{n+1}^{[48,32,0]} \oplus s_n^{[56,55,54]} = 1 \oplus C_{[4,0,6,2,7,3],n}^{[0]}. \quad (33)$$

The probability that this equation holds is about 0.5004575, i.e., the probability bias of this equation is about $\varepsilon^{max} = 0.0004575$.

## IV. The Distinguisher and its Error Probability

In this section, we construct the distinguisher by using the linear correlations between consecutive key streams $s_n$ and $s_{n+1}$.

### A. The derived distinguisher

Following the methods to construct distinguisher [9, 10], we can derive our distinguisher by using equation (33). The distinguisher is described as follows.

***Step 1*** The stream cipher is iterated $N$ times. The sequence $z_n$ is computed as

$$z_n = s_{n+1}^{[48,32,0]} \oplus s_n^{[56,55,54]} \oplus G_n, n = 1, 2, \cdots, N \quad (34)$$

where

$$G_n = \begin{cases} 1 & n = 1, 3, 5, \cdots \\ 0 & n = 2, 4, 6, \cdots \end{cases} \quad (35)$$

That is to say, the counter bits $C_{[4,0,6,2,7,3],n}^{[0]}$ is guessed to be 0 at the start $n = 1$.

***Step 2*** The number of zeros and ones in the sequence $z_n$, denoted by $N_0$ and $N_1$, is counted. Then the logarithmic likelihood ratio (*LLR*) [9] is computed as

$$\begin{aligned} &LLR(z_n) \\ &= \begin{cases} N_0 log_2 \frac{0.5+\varepsilon^{max}}{0.5} + N_1 log_2 \frac{0.5-\varepsilon^{max}}{0.5} & N_0 \geq N/2 \\ N_0 log_2 \frac{0.5-\varepsilon^{max}}{0.5} + N_1 log_2 \frac{0.5+\varepsilon^{max}}{0.5} & N_0 < N/2 \end{cases} \end{aligned} \quad (36)$$

where $\varepsilon^{max} = 0.0004575$.

***Step 3*** We decide whether the sequence is random or generated by the stream cipher by the value of $LLR(z_n)$. If $LLR(z_n) \geq 0$, then we assume the sequence is from the cipher. If $LLR(z_n) < 0$, the sequence is random.

### B. The error probability of the algorithm

The distinguisher can make two types of mistakes: it can either output $LLR(z_n) \geq 0$ when the sequence $z_n$ is random or output $LLR(z_n) < 0$ when the sequence $z_n$ is from the cipher. The two types of error probabilities are denoted by $\alpha$ and $\beta$ respectively. Then overall error probability of the distinguisher is $P_e = \frac{1}{2}(\alpha + \beta)$.

According to (36), the logarithmic likelihood ratio can be rewritten as

$$\begin{aligned} &LLR(z_n) \\ &= \begin{cases} LLR_1 = (D - B)N_0 + BN & N_0 \geq N/2 \\ LLR_2 = (B - D)N_0 + AN & N_0 < N/2 \end{cases} \end{aligned} \quad (37)$$

where $D = log_2\frac{0.5+\varepsilon^{max}}{0.5}$ and $B = log_2\frac{0.5-\varepsilon^{max}}{0.5}$. We consider the first type of mistake. In this case, the sequence $z_n$ is random, i.e., $P(z_n = 0) = P(z_n = 1) = 0.5$. By using central limit theorem, $N_0$ can be assumed to be normally distributed with mean $N/2$ and variance $N/4$ for large $N$. Then $LLR_k\,(k = 1, 2)$ can be approximately normal with mean $(D+B)N/2$ and variance $N(D-B)^2/4$. The first type of error probability $\alpha$ is

$$\begin{aligned}
&P(LLR(z_n) \geq 0)\\
=&P(LLR(z_n) \geq 0, N_0 \geq N/2)\\
&+P(LLR(z_n) \geq 0, N_0 < N/2)\\
=&P(LLR_1 \geq 0, N_0 \geq N/2) + P(LLR_2 \geq 0, N_0 < N/2)\\
=&P(LLR_1 \geq 0, LLR_1 \geq (D+B)N/2)\\
&+P(LLR_2 \geq 0, LLR_2 \geq (D+B)N/2). \qquad (38)
\end{aligned}$$

Note that $(D + B)N/2 < 0$, then

$$\alpha = P(LLR_1 \geq 0) + P(LLR_2 \geq 0) = 2P(LLR_1 \geq 0). \quad (39)$$

The second type of mistake happens when the distinguisher outputs $LLR(z_n) < 0$ for the sequence $z_n$ generated by the cipher. In this case, $z_n$ can follow two different probability distributions: $P(z_n = 1) = 0.5 - \varepsilon^{max}$ or $P(z_n = 1) = 0.5 + \varepsilon^{max}$. For the first distribution, $LLR_1$ can be approximately normal with mean $(D-B)N(0.5-\varepsilon^{max})+BN$ and variance $N(D - B)^2(0.25 - (\varepsilon^{max})^2)$, $LLR_2$ can be approximately normal with mean $(B-D)N(0.5-\varepsilon^{max})+AN$ and variance $N(D - B)^2(0.25 - (\varepsilon^{max})^2)$. With the analysis similar to equation (38), we obtain the error probability:

$$\begin{aligned}
\beta_1 &= P(LLR(z_n) < 0)\\
&= P(0 > LLR_1 \geq (D+B)N/2)\\
&\quad +P(0 > LLR_2 > (D+B)N/2). \qquad (40)
\end{aligned}$$

For the second distribution $P(z_n = 1) = 0.5 + \varepsilon^{max}$, the corresponding error probability $\beta_2$ can be computed similarly. Then the second type of error probability can be computed as $\beta = (\beta_1 + \beta_2)/2$.

TABLE II
ERROR PROBABILITY FROM THEORETICAL ANALYSIS FOR SOME SELECTED $N$.

| N | $(\varepsilon^{max})^{-2}$ | $2(\varepsilon^{max})^{-2}$ | $4(\varepsilon^{max})^{-2}$ | $8(\varepsilon^{max})^{-2}$ |
|---|---|---|---|---|
| $\alpha$ | 31.7% | 15.7% | 4.6% | 0.5% |
| $\beta$ | 15.7% | 7.9% | 2.3% | 0.2% |
| $P_e$ | 23.7% | 11.8% | 3.45% | 0.35% |

TABLE III
ACTUAL ERROR RATES OF OUR DISTINGUISHER FOR SOME SELECTED $N$.

| N | $(\varepsilon^{max})^{-2}$ | $2(\varepsilon^{max})^{-2}$ | $4(\varepsilon^{max})^{-2}$ | $8(\varepsilon^{max})^{-2}$ |
|---|---|---|---|---|
| $E_\alpha/100$ | 33/100 | 16/100 | 4/100 | 0/100 |
| $E_\beta/100$ | 17/100 | 9/100 | 2/100 | 0/100 |
| $(E_\alpha + E_\beta)/200$ | 50/200 | 25/200 | 6/200 | 0/200 |

Thus we get the overall error probability of the distinguisher $P_e(N) = \frac{1}{2}(\alpha + \beta)$. It is a function of $N$, the number of consecutive key streams used in the distinguisher. We calculate $P_e(N)$ for some selected values of $N$. The results are shown in Table 2. It can be found that the $P_e(N)$ tends to close to zero when $N$ increases. We perform experiments to ensure the effectiveness of our distinguisher. For some selected $N$, we apply our distinguisher to key streams generated randomly and by the cipher respectively. Then we decide whether the key streams tested is from the cipher or it is perfectly random. The experiment is repeated 100 times for each selected $N$. Then the times that error occurs, $E_\alpha$ and $E_\beta$, are counted. Then, the actual two types of error rates are $E_\alpha/100$ and $E_\beta/100$. The actual overall error rate is $(E_\alpha + E_\beta)$/200. The results are shown in the Table 3. It is found the actual error rates agree well with the theoretical results. When the number of consecutive key streams generated by the cipher, $N \geq 8(\varepsilon^{max})^{-2} \approx 2^{25}$, our algorithm can distinguish the cipher streams from perfectly random streams with the error rate nearly equal to 0.

## V. AN IMPROVEMENT OF THE CIPHER

In this section, we make an improvement to the original stream cipher to resist the distinguishing attack described above. As is demonstrated, since the state-update function of the cipher has good linear approximations and the counter bits are easily predicted, the cipher is susceptible to distinguishing attacks. There are two methods to improve the original stream cipher. The first one is to modify the counter system such that it is difficult to predict the counter bits. An example applying this method can be found in [12]. The improvement of the counter will not affect the encryption speed significantly. However, since the effective linear approximations of the state-update function still exist, this improvement seems unreliable. A better improvement is to iterate the state-update function several times to avoid effective linear approximations. That is to say, the state-update function is improved as

$$X_{n+1} = F^R(X_n) \oplus C_n \qquad (41)$$

where $F$ is the original state-update function excluding the counter (see Fig. 1.) and $R$ is the iterating times. Since the the best linear approximations of $F$ have the probability bias of $2^{-11}$, the probability bias of the best linear approximations of $F^R$, denoted by $\varepsilon_{F^R}$, should be larger than $\varepsilon_{F^R} = 2^{R-1}2^{-11R}$ due to Piling-Up Principle [10]. If we select $R = 3$, then $\varepsilon_{F^R} = 2^{-31}$. The number of consecutive key streams required to apply the distinguishing attack is about $q\varepsilon_{F^R}^{-2} = q2^{62}$, where $q$ is a small positive integer [9]. For the cipher, the lower bounds of the cycle length is $2^{64}$. That is to say, to avoid using key streams repeatedly, the cipher will not be iterated more than $2^{64}$ times with the same seed key in the application. Therefore, the distinguishing attack proposed above is infeasible to the improved cipher when the iterating times $R = 3$.

## VI. CONCLUSION

In conclusion, in this brief, we have applied linear cryptanalysis to a chaos-based stream cipher, which was designed by following the basic design criterion of cryptosystem – confusion and diffusion and have some desirable properties. A

distinguishing attack on the cipher is presented and this attack promotes the improvement of the cipher. Our work implies that the strict cryptographic tools should be more effectively used in the design and cryptanalysis of chaotic stream ciphers.

## REFERENCES

[1] L. Kocarev, "Chaos-based cryptography: A brief overview," IEEE Circuits Syst. Mag., vol. 1, pp. 6–21, 2001.
[2] G. Jakimoski and L. Kocarev, "Chaos and cryptography: Block encryption ciphers based on chaotic maps," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 48, no. 2, pp. 163–169, Feb. 2001.
[3] G. Jakimoski and L. Kocarev, "Differential and Linear Probabilities of a Block-Encryption Cipher", IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 1, pp. 121–123, Jan. 2003.
[4] J. M. Amigo and J. Szczepanski, "Approximations of dynamical systems and their applications to cryptography," Int. J. Bifurc. Chaos, vol. 13, pp. 1937–1948, 2003.
[5] J. Szczepanski, J. M. Amigo, T. Michalek, L. Kocarev, "Cryptographically secure substitutions based on the approximation of mixing maps," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 52, no. 2, pp. 443–453, Feb. 2005.
[6] N. Masuda, G. Jakimoski, K. Aihara, L. Kocarev, "Chaotic Block Ciphers: From Theory to Practical Algorithms," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 53, no. 6, pp. 1341–1352, Jun. 2006.
[7] R. Yin, J. Yuan, Q. Yang, et al., "Discretization of coupled map lattices for a stream cipher," submitted to Physics Letters A, Apr. 2009.
[8] S. Paul, B. Preneel, G. Sekar, "Distinguishing Attacks on the Stream Cipher Py," Fast Software Encryption 2006, M.J.B. Robshaw, ed., vol. 4047, pp. 405–421, Springer Berlin/Heidelberg 2006.
[9] T. Baigneres, P. Junod, S. Vaudenay, "How Far Can We Go Beyond Linear Cryptanalysis?," ASIACRYPT 2004, P.J. Lee, ed., vol. 3329, pp. 432–450, Springer Berlin/Heidelberg 2004.
[10] M. Matsui, "Linear cryptanalysis method for DES ciphers," Advances in Cryptology–Eurocrypt 1993, T. Helleseth, ed., vol. 765, pp. 386–397, Springer Berlin/Heidelberg 1994.
[11] M. Boesgaard, M. Vesterager, T. Pedersen, et al., "Rabbit: A New High-Performance Stream Cipher," Fast Software Encryption 2003, T. Johansson, ed., vol. 2887, pp. 307–329, Springer Berlin/Heidelberg 2003.