Cryptanalysis and improvement on a block cryptosystem based on iteration a chaotic map

Yong Wang a,b,*, Xiaofeng Liao a, Tao Xiang a, Kwok-Wo Wong c, Degang Yang a

a Department of Computer Science and Engineering, Chongqing University, Chongqing 400044, China
b Chongqing University of Posts and Telecommunications, Chongqing 400065, China
c Department of Electronic Engineering, City University of Hong Kong, Hong Kong

Received 14 May 2006; accepted 6 November 2006
Available online 20 November 2006
Communicated by A.R. Bishop

Abstract

Recently, a novel block encryption system has been proposed as an improved version of the chaotic cryptographic method based on iterating a chaotic map. In this Letter, a flaw of this cryptosystem is pointed out and a chosen plaintext attack is presented. Furthermore, a remedial improvement is suggested, which avoids the flaw while keeping all the merits of the original cryptosystem.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Cryptanalysis; Chaos; Chaotic cryptosystem; Remedial improvement

1. Introduction

Properties of chaotic systems such as ergodicity and sensitive dependence on initial conditions and system parameters are quite advantageous in constructing secure communication schemes. In the past few years, various chaos-based secure communication systems have been proposed. Since Baptista proposed a new cryptosystem based on the property of ergodicity of chaotic systems [2], some variants and their cryptanalysis have been published [3–10]. Recently, a novel block cryptosystem based on iterating a chaotic map is presented [1], which possesses the advantages such as flat ciphertext and fast encryption speed.

This block cryptosystem iterate the logistic map

\[ \tau(x) = \mu x (1 - x), \quad x \in I = [0, 1], \] (1)
to generate a sequence of independent and identically distributed random variables. Expressing the value of \( x \) in binary representation, we have

\[ x = b_1(x) b_2(x) \cdots b_l(x) \cdots, \quad x \in [0, 1], \quad b_l(x) \in \{0, 1\}. \] (2)

As a result, a binary sequence \( B^n_i = \{b_l(\tau^n(x))\}_{n=0}^{\infty} \) where \( n \) is the length of the sequence and \( \tau^n(x) \) is the \( n \)th iteration of the logistic map, can be obtained. The scheme is described below and an illustration is given in Fig. 1.

Step 1. Get the start point \( \omega \) which denotes the real value of \( x \) from the last \( N_0 \) transient iterations, i.e., \( \omega = \tau^{N_0}(x_0) \).

Step 2. Divide the message \( m \) into subsequences \( \omega_j \) of length \( l \) bytes (here \( l = 8 \)):

\[ m = p_0 p_1 \ldots p_{l-1}, p_l p_{l+1} \ldots p_{2l-1}, p_{2l} \ldots. \] (3)

Get eight bytes of plaintext \( p_j, p_{j+1}, \ldots, p_{j+7} \) and combine them to form a binary message block \( P_j \) with 64 bits \( P_j = p_j p_{j+1} \ldots p_{j+7} \).

Step 3. Based on the method to generate binary sequences by iterating the logistic map, obtain the binary sequences \( A_j = B_1^i B_2^i \ldots B_{64}^i \) and \( A_j' = B_1^{i+1} B_2^{i+1} \ldots B_{70}^{i+1} \) formed by all the third bits, i.e., \( i = 3 \) in Eq. (2), through 70 iterations of the logistic map. An integer \( D_j \) is computed as the decimal value of \( A_j' \).
This value will be used to iterate the logistic map successively for \( D_j \) times after the current block has been encrypted.

Step 4. Permute the message block \( P_j \) with left cyclic shift \( D_j \) bits to obtain a new message block \( P'_j \) according to the approach illustrated in Fig. 2.

Step 5. Perform the following manipulation with sequences \( P'_{j} \) and \( A_j \):

\[
C_j = P'_{j} \oplus A_j
\]

(4)

where \( \oplus \) denotes the XOR operation. As a result, the ciphertext block \( C_j \) for the message block \( P_j \) is obtained. Dividing the ciphertext block \( C_j \) into 8-bit partitions and we obtain the ciphertext \( c_j, c_{j+1}, \ldots, c_{j+7} \) of the plaintext bytes \( p_j, p_{j+1}, \ldots, p_{j+7} \), respectively.

Step 6. If all the plaintexts have already been encrypted, the encryption process is finished. Otherwise, let \( \omega = \tau^{70+D_j}(\omega) \) and go to Step 2.

For a detail explanation of this cryptosystem, it is highly recommended the thorough reading of [1]. The remaining of the Letter is organized as follows. A flaw of this cryptosystem is pointed out in Section 2 while a chosen plaintext attack is presented in Section 3. A remedial improvement is suggested in Section 4. Finally, conclusions are drawn in Section 5.

2. A flaw of the cryptosystem

Although the cipher proposed in [1] might look like a block cipher, it behaves as a stream cipher indeed [11, p. 20]. The operation of the above encryption scheme as a stream cipher can be explained as follows. Suppose \( P = P_1 P_2 \ldots \) is the plaintext and \( \mu, x_0 \) is the secret key, the logistic maps are used to generate a keystream \( K = (D_1 A_1)(D_2 A_2) \cdot \cdot \cdot \). This keystream is used to encrypt the plaintext according to the rule:

\[
C = C_1 C_2 \cdot \cdot \cdot = E_{D_1 A_1}(P_1) E_{D_2 A_2}(P_2) \cdot \cdot \cdot .
\]

(5)

Decrypting the ciphertext \( C \) can be accomplished by computing the keystream. From Figs. 1 and 2, we observe that \( D_j \) and \( A_j \) are independent of the plaintext and is determined only by the secret key \( (\mu, x_0) \). This is the main flaw of the cryptosystem. To demonstrate the security loophole caused by this flaw, we first encrypt two different plaintext sequences. They are ‘abcdefghijklmnopqrstuvwxyz123456...’ and ‘0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ’. The secret keys are set to \( \mu = 3.99999995 \) and \( x_0 = 0.1777 \). We fill each block of plaintext \( P_j, D_j, A_j \) and the corresponding ciphertext \( C_j \) into Tables 1 and 2, respectively. Note: in this Letter, data in all tables are expressed in hexadecimal format.

From Tables 1 and 2, we observe that the sequences in columns \( D_j \) and \( A_j \) are totally identical. This is not caused by coincidence, but resulted from the flaw mentioned above. Once \( \mu \) and \( x_0 \) are determined, the sequences \( D_j \) and \( A_j \) are fixed too, regardless of the plaintext.

3. The attack

3.1. Classical attacks

One measure of the security of a cryptosystem is its resistance to standard cryptanalysis under the assumption that the cryptanalyst has the details of the algorithm. This is an evident requirement in today’s secure communications networks, usually referred to as Kerchoff’s principle [11, p. 24]. Cryptanalysis is the study of taking encrypted data, and trying to decrypt it without use of the key. There are numerous techniques for performing cryptanalysis, depending on what access the cryptanalyst has to the plaintext, ciphertext, or other aspects of the cryptosystem. Below are some of the most common types of attacks, ordered from the hardest type of attack to the easiest:

![Fig. 1. Block diagram of the scheme.](image1)

![Fig. 2. Detail description of permutation.](image2)

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption of the plaintext ‘abcdefghijklmnopqrstuvwxyz123456...’ using ( \mu = 3.9999995, \ x_0 = 0.1777 )</td>
</tr>
<tr>
<td>( j )</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
Table 2

Encryption of the plaintext '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ' using $\mu = 3.9999995$, $x_0 = 0.1777$

<table>
<thead>
<tr>
<th>$j$</th>
<th>$P_j$</th>
<th>$D_j$</th>
<th>$A_j$</th>
<th>$C_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30,31,32,33,34,35,36,37</td>
<td>1d</td>
<td>b2,93,08,6d,8d,93,50,cf</td>
<td>d4,15,ae,ab,6b,95,76,89</td>
</tr>
<tr>
<td>2</td>
<td>38,39,41,42,43,44,45,46</td>
<td>08</td>
<td>e1,ed,d3,bb,8b,27,be,b8</td>
<td>d8,ac,91,f8,cf,62,f8,80</td>
</tr>
<tr>
<td>3</td>
<td>47,48,49,4a,4b,4c,4d,4e</td>
<td>19</td>
<td>71,b9,86,3d,81,a4,d8,8d</td>
<td>e5,2f,1e,a7,1d,2a,48,1d</td>
</tr>
<tr>
<td>4</td>
<td>4f,50,51,52,53,54,55,56</td>
<td>20</td>
<td>9e,ad,e4,c9,7b,39,e7,e9</td>
<td>cd,f9,b1,9f,34,69,b6,bb</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3. The flow chart of chosen text attack.

(1) Ciphertext-only attack: the cryptanalyst has a piece of ciphertext, with no plaintext. This is probably the most difficult type of cryptanalysis, and calls for a bit of guesswork.

(2) Known-plaintext attack: the cryptanalyst has knowledge of a portion of the plaintext from the ciphertext. Using this information, the cryptanalyst attempts to deduce the key used to produce the ciphertext. The cryptanalyst has both a piece of ciphertext and the corresponding piece of plaintext.

(3) Chosen-plaintext attack: the cryptanalyst is able to have any plaintext encrypted with a key and obtain the resulting ciphertext, but the key itself cannot be analyzed. The cryptanalyst attempts to deduce the key by comparing the entire ciphertext with the original plaintext.

(4) Chosen-ciphertext attack: the cryptanalyst can cause the receiver to decrypt a ciphertext that he chose. Hence he can choose a ciphertext, and construct the corresponding plaintext.

As long as the key is figured out through one of the attacks mentioned above, the cryptosystem is considered to be insure [9].

3.2. Chosen plaintext attack

Our chosen plaintext attack is described below and an illustration is given in Fig. 3.

Step 1. Choose a special plaintext $P_z$ composed of all zeros, whose length is identical to that of the ciphertext we want to attack. Suppose $P_{zj}$ (64-bit) is the $j$th block in $P_z$ and $C_{zj}$ is the corresponding ciphertext of $P_{zj}$. We have

$$C_{zj} = F(P_{zj}, D_j) \oplus A_j$$ (6)

where function $F(A, B)$ indicates permuting $A$ by a left cyclic shift $B$ bits, according to the approach illustrated in Fig. 2. As $P_{zj}$ is composed of all zeros, $F(P_{zj}, D_j)$ is equal to $P_{zj}$. Thus we have

$$C_{zj} = F(P_{zj}, D_j) \oplus A_j = P_{zj} \oplus A_j = A_j.$$ (7)

Namely the sequence $C_{z1}, C_{z2}, \ldots$ is the same as $A_1, A_2, \ldots$. We fill in $P_{zj}, D_j, A_j$ and the corresponding ciphertext $C_{zj}$ into Table 3.

Step 2. Choose another special plaintext $P_{s}$, whose length is also the same as that of the ciphertext. Each block (64-bit) in $P_s$ is $011 \ldots 11$ in binary representation, i.e., only the first bit is zero, and the remaining bits are all one. Similarly, suppose $P_{sj}$ is the $j$th block in $P_s$, $C_{sj}$ is corresponding ciphertext of $P_{sj}$. We have

$$C_{sj} = F(P_{sj}, D_j) \oplus A_j.$$ (8)

Since the sequence $A_j$ has already been obtained from Step 1, we get:

$$C_{sj} \oplus A_j = F(P_{sj}, D_j) \oplus A_j \oplus A_j = F(P_{sj}, D_j).$$ (9)

There is only one zero in the $F(P_{sj}, D_j)$ and we can find out its position easily. Suppose $y_j$ is the position of zero in $F(P_{sj}, D_j)$ counting from rightmost bit. If $P_{sj}$ is permuted by a left cyclic shift of $y_j$ bits, we obtain $F(P_{sj}, y_j)$. Obviously, $F(P_{sj}, y_j)$ is equal to $F(P_{sj}, D_j)$. Therefore $y_j$ is equal to $D_j$. Finally we obtain the sequence $D_1, D_2, \ldots$ and fill in $P_{sj}, F(P_{sj}, D_j), D_j, y_j$ and the corresponding ciphertext $C_{sj}$ into Table 4.
Table 3
Encryption of the chosen plaintext '00000000...' using $\mu = 3.9999995$, $x_0 = 0.1777$

<table>
<thead>
<tr>
<th>$j$</th>
<th>$P_{z_j}$</th>
<th>$D_j$</th>
<th>$A_j$</th>
<th>$C_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00,00,00,00,00,00,00,00</td>
<td>1d</td>
<td>b2,93,08,6d,8d,93,50,cf</td>
<td>b2,93,08,6d,8d,93,50,cf</td>
</tr>
<tr>
<td>2</td>
<td>00,00,00,00,00,00,00,00</td>
<td>08</td>
<td>e1,ed,d3,bb,27,be,88</td>
<td>e1,ed,d3,bb,27,be,88</td>
</tr>
<tr>
<td>3</td>
<td>00,00,00,00,00,00,00,00</td>
<td>19</td>
<td>71,b9,86,3d,81,4d,8d,8f</td>
<td>71,b9,86,3d,81,4d,8d,8f</td>
</tr>
<tr>
<td>4</td>
<td>00,00,00,00,00,00,00,00</td>
<td>20</td>
<td>9e,ad,e4,97,39,c7,e9</td>
<td>9e,ad,e4,97,39,c7,e9</td>
</tr>
</tbody>
</table>

Table 4
Encryption of the chosen plaintext ‘7FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF...’ (in hexadecimal format) using $\mu = 3.9999995$, $x_0 = 0.1777$

<table>
<thead>
<tr>
<th>$j$</th>
<th>$P_{s_j}$</th>
<th>$F(P_{s_j}, D_j)$</th>
<th>$D_j$</th>
<th>$y_j$</th>
<th>$C_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7f,ff,ff,ff,ff,ff,ff,ff</td>
<td>ff,ff,ff,ff,ff,ff,ff,ff</td>
<td>1d</td>
<td>1d</td>
<td>4d,6c,ef,79,26,6c,af,30</td>
</tr>
<tr>
<td>2</td>
<td>7f,ff,ff,ff,ff,ff,ff,ff</td>
<td>ff,ff,ff,ff,ff,ff,ff,7f</td>
<td>08</td>
<td>08</td>
<td>1e,12,2c,44,74,d8,41,c7</td>
</tr>
<tr>
<td>3</td>
<td>7f,ff,ff,ff,ff,ff,ff,ff</td>
<td>ff,ff,ff,ff,ff,ff,ff,ff</td>
<td>19</td>
<td>19</td>
<td>8e,46,79,c2,7f,5b,27,70</td>
</tr>
<tr>
<td>4</td>
<td>7f,ff,ff,ff,ff,ff,ff,ff</td>
<td>ff,ff,ff,ff,ff,ff,ff,ff</td>
<td>20</td>
<td>20</td>
<td>61,52,1b,36,04,c6,18,16</td>
</tr>
</tbody>
</table>

Fig. 4. (a) Plaintext, (b) the ciphertext, (c) the result of attack.

By following this computationally inexpensive method, we can obtain as many $A_j$ and $D_j$ as desired, which is equivalent to knowing the secret key.

3.3. The result of attack

We encrypt a bitmap image file named Lena with the scheme in [1] and break it by our approach of chosen plaintext attack. The original image, the encrypted image and the image recovered from the attack is shown in Figs. 4(a)–(c), respectively.

4. A remedial improvement

The plaintext-independent keystream $D$ and $A$ cause the encryption algorithm presented in [1] vulnerable to chosen plaintext attack. Except this flaw, the cryptosystem is excellent in confusion, diffusion as well as efficiency. Therefore it is valuable to propose a modified cryptosystem to get rid of this flaw. A straightforward improvement is to make the value of $D_j$ and $A_j$ dependent on both the key and the plaintext. Because the ciphertext is related to the plaintext, we use it as a feedback to change the value of $D_j$. Details of the improvement are listed below:

The first four steps and the 6th step are the same as the encryption algorithm presented in [1]. In Step 5, after $C_j$ has been generated, we introduce two equations:

$$f(C_j) = c_1 + c_2 + \cdots + c_8, \quad (10)$$

$$D^* = D + f(C_j) \mod 64, \quad (11)$$

where $C_j$ is the $j$th ciphertext block and $c_i$ ($i = 1, 2, \ldots, 8$) is the $i$th byte of $C_j$. We iterate the logistic map $D^*$ times instead of $D$ times.

Obviously, different plaintext sequences result in different ciphertext blocks, and eventually different sequences of $D_j$ and $A_j$. The difference between the columns $D_j$ and $A_j$ in Tables 5 and 6 illustrates the effectiveness of this remedy.

Both Eqs. (10) and (11) involve only simple computation. Therefore, our remedy is not a time-consuming operation and will not lead to a substantial loss of efficiency comparing with the original cryptosystem. As far as the security is concerned, Figs. 5 and 6 show that the modified cryptosystem has enhanced the diffusion property.

5. Conclusion

According to Refs. [2–11], two important things should be considered when designing a chaotic cryptosystem. The first one is that the distribution of the ciphertext should be sufficiently flat in order to resist the statistics attack. The other is that the subkeys should depend on not only the secret key but
Table 5
Encryption of the plaintext 'abcdefghijklmnopqrstuvwxyz123456...' with the modified cryptosystem using $\mu = 3.9999995$, $x_0 = 0.1777$

<table>
<thead>
<tr>
<th>$j$</th>
<th>$P_j$</th>
<th>$D_j$</th>
<th>$A_j$</th>
<th>$C_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>61,62,63,64,65,66,67,68</td>
<td>1d</td>
<td>b2,93,08,6d,8d,93,50,cf</td>
<td>76,55,c0,a7,41,5d,80,0d</td>
</tr>
<tr>
<td>2</td>
<td>69,6a,6b,6c,6d,6e,6f,70</td>
<td>30</td>
<td>71,64,f7,d7,04,37,38,dc</td>
<td>18,0e,9c,69,59,57,ac</td>
</tr>
<tr>
<td>3</td>
<td>71,72,73,74,75,76,77,78</td>
<td>0e</td>
<td>1d,8c,43,9e,ad,e4,c9,7b</td>
<td>0a,ab,74,d9,fa,83,be,fc</td>
</tr>
<tr>
<td>4</td>
<td>79,7a,31,32,33,34,35,36</td>
<td>34</td>
<td>fd,30,51,7a,63,6c,d9,e9</td>
<td>cc,02,62,4e,56,5a,a0,93</td>
</tr>
</tbody>
</table>

Table 6
Encryption of the plaintext '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ...' with the modified cryptosystem using $\mu = 3.9999995$, $x_0 = 0.1777$

<table>
<thead>
<tr>
<th>$j$</th>
<th>$P_j$</th>
<th>$D_j$</th>
<th>$A_j$</th>
<th>$C_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30,31,32,33,34,35,36,37</td>
<td>1d</td>
<td>b2,93,08,6d,8d,93,50,cf</td>
<td>d0,7f,6e,05,ef,3f,ae,af</td>
</tr>
<tr>
<td>2</td>
<td>38,39,41,42,43,44,45,46</td>
<td>0e</td>
<td>ba,77,71,64,f7,d7,04,37</td>
<td>39,e3,65,40,c3,93,50,54</td>
</tr>
<tr>
<td>3</td>
<td>47,48,49,4a,4b,4c,4d,4e</td>
<td>31</td>
<td>61,8f,60,69,36,23,d9,8e</td>
<td>f7,17,fa,fa,b3,4b,1a</td>
</tr>
<tr>
<td>4</td>
<td>4f,50,51,52,53,54,55,56</td>
<td>1d</td>
<td>3f,4c,14,5e,98,db,36,7a</td>
<td>9f,ee,b0,f8,30,71,9a,e4</td>
</tr>
</tbody>
</table>

Fig. 5. Difference in ciphertext when encrypting two plaintext sequences with only the first bit different based on the cryptosystem proposed in [1].

Fig. 6. Difference in ciphertext when encrypting two plaintext sequences with only the first bit different based on this modified cryptosystem.

also the plaintext to avoid keystream attack. Except for this flaw, the cryptosystem proposed in [1] is excellent in confusion, diffusion and efficiency.

The modified cryptosystem described in Section 4 keeps all these excellent properties while avoids the flaw. Therefore, it is suitable for practical use such as the secure transmission of confidential information over the Internet.

Acknowledgements

The work described in this Letter was supported by the National Natural Science Foundation of China (No. 60271019), the Doctorate Foundation Grants from the Ministry of Education of China (No. 20020611007), the Post-doctoral Science Foundation of China and the Natural Science Foundation of Chongqing (No. 8509).

References