Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2010, Article ID 590590, 14 pages
doi:10.1155/2010/590590

Research Article

Cryptanalysis and Improvement of a Block Cipher Based on Multiple Chaotic Systems

Jun He,[1] Haifeng Qian,[1] Yuan Zhou,[2] and Zhibin Li[1]

[1] *Department of Computer Science, East China Normal University, Shanghai 200241, China*
[2] *National Computer Network Emergency Response Technical Team (Coordination Center of China), Beijing 100029, China*

Correspondence should be addressed to Haifeng Qian, hfqian@cs.ecnu.edu.cn

Received 31 January 2010; Accepted 15 April 2010

Academic Editor: Ming Li

Wang and Yu proposed a block cipher scheme based on dynamic sequences generated by multiple chaotic systems, which overcomes the problem of periodical degradation on random sequences due to computational precision. Their scheme has a feature that a plaintext is encrypted by a keystream created from several one-dimensional chaotic maps. However, this feature results in some weaknesses of the encryption algorithm. We show three kinds of attacks in this paper, through which one can recover the plaintext from a given ciphertext without the secret key. We also present an improvement on their scheme, which prevents the three attacks mentioned above. Security of the enhanced cipher is presented and analyzed, which shows that our improved scheme is secure under the current attacks.

1. Introduction

The chaos-based encryption scheme was first proposed in 1989 [1]. Following the work, a lot of cryptography researchers have proposed many chaos-based encryption schemes (some of them are the improvements on the previous ones) [2–6]. Security of all these schemes relies on the properties of chaotic systems: the sensitive dependence on initial conditions and system parameters, pseudorandom property, nonperiodicity and topological transitivity.

There are two types of cipher schemes in the chaos-based cryptosystems: stream ciphers and block ciphers. In the chaotic stream ciphers [2, 7–9], a pseudorandom sequence is generated by chaotic sequence generator to encrypt the plaintext. However, the limited computational precision degrades the pseudorandom sequence to a periodic sequence eventually. The chaotic block ciphers adopt chaotic maps to generate parameters used in encryption and decryption procedures. Pareek has proposed two block ciphers based on external keys [10, 11]. But too much time consumption in computation makes them hard to implement in the real-time telecommunication.

Wang and Yu proposed a new block encryption scheme in [12] (the Wang-Yu scheme) by combining these two methods of chaotic cryptography. Their scheme provides not only good randomness but also high computational efficiency. In their scheme, several one-dimensional chaotic maps are used to generate pseudorandom sequences with independent and uniform distribution. Through a series of transformations, the sequences constitute a keystream randomly distributed in the key space. The keysteam is used to encrypt the plaintext by executing simple operations such as Exclusive-OR (XOR) and shifting repeatedly with sufficient rounds.

Generally speaking, a secure cipher is supposed to resist the following attacks: the chosen plaintext attacks (CPAs), the chosen ciphertext attacks (CCAs) and the known plaintext attacks (KPAs). Unfortunately, the Wang-Yu cipher cannot resist any of the above attacks because the keystream remains unchanged during the execution of the encryption procedure each time. Thus, for an attacker, knowing the keystream is equivalent to knowing the secret key.

Our Contributions

We point out the drawbacks of the Wang-Yu block cipher based on dynamic sequences generated by multiple chaotic systems. Their scheme is vulnerable to the following three kinds of attacks: the chosen plaintext attacks, the chosen ciphertext attacks, and the known plaintext attacks. In order to obtain a secure block cipher from chaotic systems, we make efforts to improve the Wang-Yu block cipher. We design a new block cipher which makes the keystream sensitive to any change of the plaintext and the ciphertext. Therefore, our new block cipher is able to resist the above attacks. On the other hand, our scheme preserves the high computational efficiency of the original one.

In Section 2, some essential notations and security definitions are introduced. In Section 3, the Wang-Yu block cipher is reviewed. In Section 4, we analyze the Wang-Yu scheme and show three different attacks to the scheme. In Section 5, an improved block cipher scheme is proposed. We analyze security and discuss the efficiency of the new scheme in Section 6. Finally, we conclude the paper in Section 7.

2. Preliminaries

2.1. Notations

We use the following notations:

P: Plaintext,

P_i: ith plaintext block,

C: Ciphertext,

C_i: ith ciphertext block,

L: Number of plaintext blocks,

r: Number of transformation rounds,

$c(l)$: Index of a chosen chaotic map,

$B_i^{(l)}$: 64-bit temporary value in encryption/decryption transformation,

$S_i^{c(l)}$: 64-bit value generated by a chaotic map in the Wang-Yu scheme,

S_i: Keystream of ith plaintext block in the Wang-Yu scheme,

$SK_i^{c(l)}$: 64-bit value generated by a chaotic map in the improved scheme,

SK_i: Key stream of ith plaintext block in the improved scheme,

x_j: A real number in $(0,1)$ indexed by j.

2.2. Definitions and Security Notions

We review the definition of chaos and security notions for block cipher as follows.

Definition 2.1 (Chaos). Chaos is aperiodic time-asymptotic behaviour in a deterministic system which exhibits sensitive dependence on initial conditions.

This definition contains three main elements.

(1) *Aperiodic time-asymptotic behaviour*: this implies the existence of phase-space trajectories which do not settle down to fixed points or periodic orbits. For practical reasons, we insist that these trajectories are not too rare. We also require the trajectories to be bounded, that is, they should not go off to infinity.

(2) *Deterministic*: this implies that the equations of motion of the system possess no random inputs. In other words, the irregular behaviour of the system arises from nonlinear dynamics and not from noisy driving forces.

(3) *Sensitive dependence on initial conditions*: this implies that nearby trajectories in phase-space separate exponentially fast in time; that is, the system has a positive Lyapunov exponent.

Definition 2.2 (One-way function). A function $f : \{0,1\}^* \to \{0,1\}^*$ is one-way if $f(\cdot)$ can be computed by a polynomial time algorithm, but for every randomized polynomial time algorithm $\mathcal{A}$,

$$\Pr\left[f\left(\mathcal{A}(f(x))\right) = f(x)\right] < \frac{1}{p(n)}, \tag{2.1}$$

for every polynomial $p(n)$ and sufficiently large n, assuming that x is chosen from the uniform distribution on $\{0,1\}^n$.

Definition 2.3 (Block cipher). A symmetric key block cipher consists of two PPT algorithms $(E_k(\cdot), D_k(\cdot))$ with the following properties: for any random $k \in_R \{0,1\}^\kappa$, the encryption algorithm on input $m \in \{0,1\}^n$ and k, outputs a ciphertext $c = E_k(m)$; the decryption algorithm on input c and k, outputs a plaintext m if $c = E_k(m)$. For any $k \in_R \{0,1\}^\kappa$ and $m \in \{0,1\}^n$, correctness requires the following to be hold:

$$m = D_k(E_k(m)). \tag{2.2}$$

Definition 2.4 (One-way CPA). Let $E_k(\cdot)$ be a block cipher. If any adversary $\mathcal{A}$ (any PPT algorithm) that is allowed to obtain the ciphertext of any message, cannot extract the plaintext from a challenge ciphertext, we say that $E_k(\cdot)$ is one-way under the chosen plaintext attacks.

Definition 2.5 (One-way CCA). Let $E_k(\cdot)$ be a block cipher. If any adversary $\mathcal{A}$ (any PPT algorithm) that is allowed to obtain the plaintext of any ciphertext (except for the challenge ciphertext), cannot extract the plaintext from a challenge ciphertext, we say that $E_k(\cdot)$ is one-way under the chosen ciphertext attacks.

Definition 2.6 (One-way KPA). Let $E_k(\cdot)$ be a block cipher. If any adversary $\mathcal{A}$ (any PPT algorithm) that is given a set of random plaintexts and corresponding ciphertexts (except for the challenge ciphertext), cannot extract the plaintext from a challenge ciphertext. We say that $E_k(\cdot)$ is one-way under the known plaintext attacks.

3. Review of the Wang-Yu Block Cipher

3.1. Algorithm Description

In this section, we briefly review the block cipher proposed by Wang and Yu [12]. In their scheme, plaintext blocks are converted into ciphertext blocks after several round transformations with XOR and shift operations. A number of 64-bit binary strings as the keystreams are generated in such transformations.

Let us see how the keystream is generated. There are two tables in the Wang-Yu scheme. One table consists of four one-dimensional chaotic maps. The other called chaotic map set (CMS) includes initial values between 0 and 1, which are produced through a random number generator from a given secret key. At the beginning, one map is randomly chosen from the first table. An initial value is also chosen from the CMS table by certain rules. The chosen map is then iterated with the initial value for 64 times. Each time the map generates a new real number. If the new number is bigger than 0.5, we get 1 for the corresponding digit. Otherwise, we get 0. Eventually, we get a 64-bit binary string after 64 iterations.

3.2. Procedure in Detail

The Wang-Yu scheme is described as follows.

(i) Encryption of $P_i \in \{0,1\}^{64}$: $C_i = E_k(P_i)$.

(1) Initialization: $B_i^{(0)} = P_i$; $l = 1$; $d = 1$; $r \in_R \{0,1\}^*$; CMS table$\leftarrow k$.
Here, $k \in_R \{0,1\}^{64}$ is the secret key of the block cipher.

(2) $c(l) \in_R [0,3]$,
$x_0 \leftarrow$ value of cth column, dth row in the CMS table.

(3) For $j = 1$ to 64:

(a) if $c(l) = 0$: $x_j = \mu x_{j-1}(1 - x_{j-1})$;
(b) if $c(l) = 1$: $x_j = \mu \sin(\pi x_{j-1})$;
(c) if $c(l) = 2$: $x_j = \mu \cos(\pi |x_{j-1} - 0.5|)$;
(d) if $c(l) = 3$: $x_j = 1 - \mu |x_{j-1} - 0.5|$.

(4) For $j = 1$ to 64:

$$s_j = \begin{cases} 1, & x_j \geqslant \overline{x}, \\ 0, & x_j < \overline{x}, \end{cases} \tag{3.1}$$

where $\overline{x} = 0.5$. The keystream is $S_i^{c(l)} = s_1 s_2 \cdots s_{64}$.

(5) Encryption transformation:

$$\begin{aligned} B_i^{(2l-1)} &= B_i^{(2l-2)} \oplus S_i^{c(l)}, \\ B_i^{(2l)} &= B_i^{(2l-1)} \ll 16\,\text{bits}. \end{aligned} \tag{3.2}$$

(6) If $l = r$, go to step (7);
else $l \leftarrow l + 1$; $d \leftarrow d + 1$; goto step (2).

(7) $C_i = B_i^{(2l)}$.
Here, the operation "$x \ll y$" represents a cyclic left shift of x by y bits.

(ii) Decryption of $C_i \in \{0,1\}^{64}$: $P_i = D_k(C_i)$.

Parameter and keystream generations here are the same as those in the encryption. The only difference is that the equations in step (5) should be replaced by

$$\begin{aligned} B_i^{(2l-1)} &= B_i^{(2l)} \gg 16\,\text{bits}, \\ B_i^{(2l-2)} &= B_i^{(2l-1)} \oplus S_i^{c(l)}. \end{aligned} \tag{3.3}$$

Here, the operation "$x \gg y$" represents a cyclic right shift of x by y bits.

3.3. Weaknesses of the Scheme

A keystream $S_i = (S_i^{c(1)}, S_i^{c(2)}, \ldots, S_i^{c(r)})$ in the Wang-Yu scheme is generated by a certain secret key k. Then it is used to encrypt the plaintext according to the following rule:

$$C_i = E_{S_i}(P_i), \quad \text{for } i = 1, \ldots, L. \tag{3.4}$$

Decryption of a ciphertext block C_i can be accomplished by calculating the corresponding keystream S_i if the key is given and doing the reverse operations of encryption $E_{S_i}(\cdot)$. However, the block cipher is not secure because some problems occur in their keystream generation and the encryption algorithm. Exactly, if we know the keystream S_i, we can recover the plaintext of a given ciphertext without the secret key.

In the next section we will show how to recover the keystream under the chosen plaintext attack, the chosen ciphertext attack and the known plaintext attack, respectively. We note that knowing the keystream S_i generated by a certain secret key is equivalent to knowing the key indeed [3].

4. Cryptanalysis of the Wang-Yu Block Cipher

With the help of the keystream S_i, we can recover the plaintext from a given ciphertext. Therefore, the following attacks focus on recovering the keystream S_i. Suppose that we have a challenge ciphertext C composed by C_i $(i = 1, 2, \ldots, L)$ to "decrypt" without the secret key. We shall calculate the keystream S_i by launching one of the attacks described later.

The encryption transformation can be described as follows:

$$\begin{aligned}
B_i^{(0)} &= P_i, \\
B_i^{(1)} &= B_i^{(0)} \oplus S_i^{c(1)}, \\
B_i^{(2)} &= B_i^{(1)} \ll 16\,\text{bits}, \\
B_i^{(3)} &= B_i^{(2)} \oplus S_i^{c(2)}, \\
B_i^{(4)} &= B_i^{(3)} \ll 16\,\text{bits}, \\
&\vdots \\
B_i^{(2r-1)} &= B_i^{(2r-2)} \oplus S_i^{c(r)}, \\
B_i^{(2r)} &= B_i^{(2r-1)} \ll 16\,\text{bits}, \\
B_i^{(2r)} &= C_i.
\end{aligned} \tag{4.1}$$

For simplicity, the encryption procedure is described as follows:

$$C_i = E_{S_i}(P_i) = (P_i \ll 16\,\text{bits})^r \oplus S_i, \quad \text{for } i = 1, \ldots, L, \tag{4.2}$$

where $S_i = ((((S_i^{c(1)} \ll 16\,\text{bits}) \oplus S_i^{c(2)}) \ll 16\,\text{bits}) \cdots \oplus S_i^{c(r)}) \ll 16\,\text{bits}$. The operation $(f)^r$ represents that the action of f is repeated r times.

Since the keystream does not change for every plaintext blocks, we can get it from a given plaintext block and a corresponding ciphertext block. Then we can use it to recover the plaintexts from other ciphertexts. A plaintext block P_i can be recovered by using the known keystream S_i and a given ciphertext block C_i as follows:

$$P_i = D_{S_i}(C_i) = \left(\left(S_i \oplus C_i\right) \gg 16\,\text{bits}\right)^r, \quad \text{for } i = 1, \ldots, L. \tag{4.3}$$

Then, we can recover the plaintexts without the secret key. The following explains how to recover the plaintexts under three different attacks.

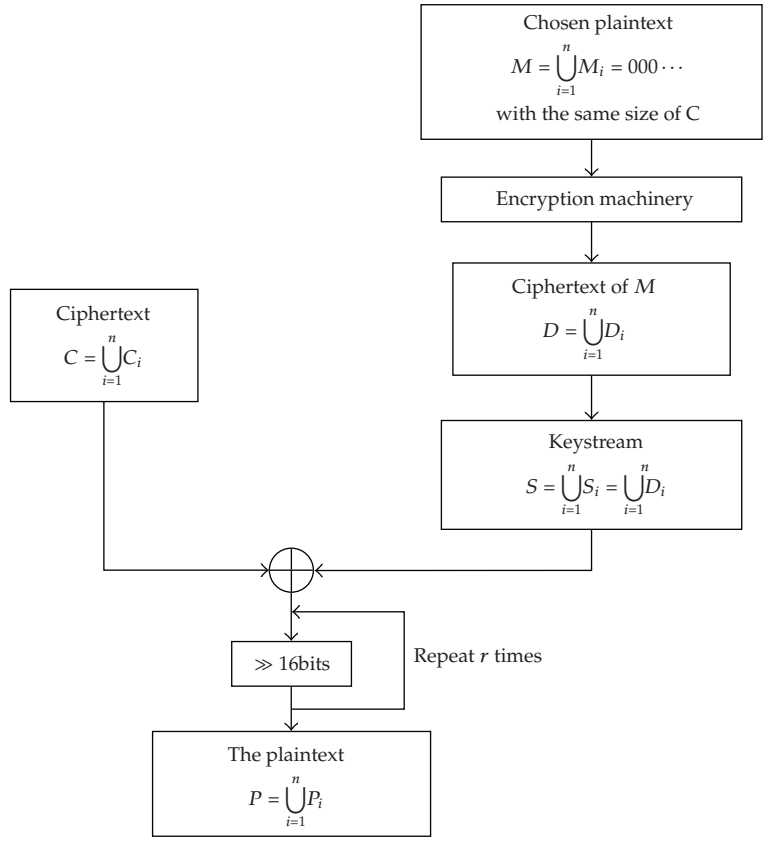


Figure 1: Flowchart of chosen plaintext attacks.

4.1. How to Recover the Plaintext under CPA

Suppose that we have obtained temporary access to the encryption machine. Given index i and a special plaintext block M_i, where $M_i = (000\cdots0)_{64}$(the ciphertext block also consists of 64 bits), we can obtain the ciphertext block D_i of the plaintext block $M_i = (000\cdots0)_{64}$ from the encryption machine.

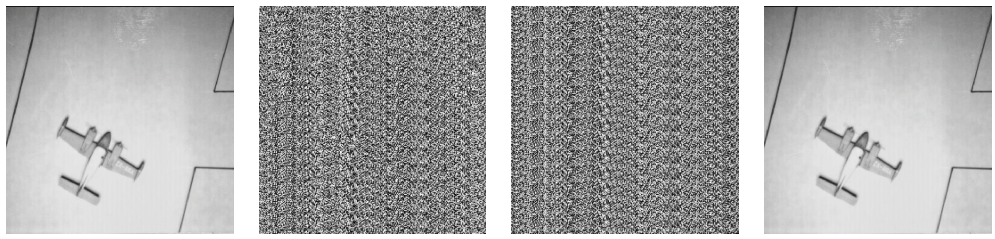
So, the keystream S_i can be generated from M_i and D_i:

$$S_i = D_i \oplus (M_i \ll 16\,\text{bits})^r = D_i, \quad \text{for } i = 1, 2, \ldots, L. \tag{4.4}$$

The recovered plaintext block can be obtained using the keystream S_i and the ciphertext block C_i as follows:

$$P_i = \Big((S_i \oplus C_i) \gg 16\,\text{bits}\Big)^r, \quad \text{for } i = 1, 2, \ldots, L. \tag{4.5}$$

The flowchart of this attack is given in Figure 1, and Figure 2 shows the simulation results of the chosen plaintext attack on a ciphered image of size 256×256.



(a) (b) (c) (d)

Figure 2: Results of the CPA: (a) the original image P; (b) ciphered image C of P; (c) ciphered image of $M = 0000$; (d) recovered image.

4.2. How to Recover the Plaintext under CCA

Assume that we have obtained temporary access to the decryption machine. Given index i and a special ciphertext block D_i, where $D_i = (000\cdots0)_{64}$(the challenge ciphertext block also consists of 64 bits), we can obtain the plaintext block M_i of the ciphertext block $D_i = (000\cdots0)_{64}$ from the decryption machine. The keystream S_i can be generated from M_i and D_i by

$$S_i = D_i \oplus (M_i \ll 16\,\text{bits})^r = (M_i \ll 16\,\text{bits})^r, \quad \text{for } i = 1, 2, \ldots, L. \tag{4.6}$$

The recovered plaintext block can be obtained by using the keystream S_i and the ciphertext block C_i as follows:

$$P_i = \Big((S_i \oplus C_i) \gg 16\,\text{bits}\Big)^r, \quad \text{for } i = 1, 2, \ldots, L. \tag{4.7}$$

The flowchart of this attack is given in Figure 3. Simulation results of a chosen ciphertext attack on a ciphered image of size 512×512 are given in Figure 4.

4.3. How to Recover the Plaintext under KPA

The knowledge of one plaintext block and its corresponding ciphertext block with the same length leads to potential damage of privacy for the cryptosystems. We know that given a plaintext block M_i and its corresponding ciphertext block D_i, the keystream can be computed as follows:

$$S_i = (M_i \ll 16\,\text{bits})^r \oplus D_i, \quad \text{for } i = 1, 2, \ldots, L. \tag{4.8}$$

The recovered plaintext block can be obtained by using the keystream S_i and the ciphertext block C_i as follows:

$$P_i = \Big((S_i \oplus C_i) \gg 16\,\text{bits}\Big)^r, \quad \text{for } i = 1, 2, \ldots, L. \tag{4.9}$$

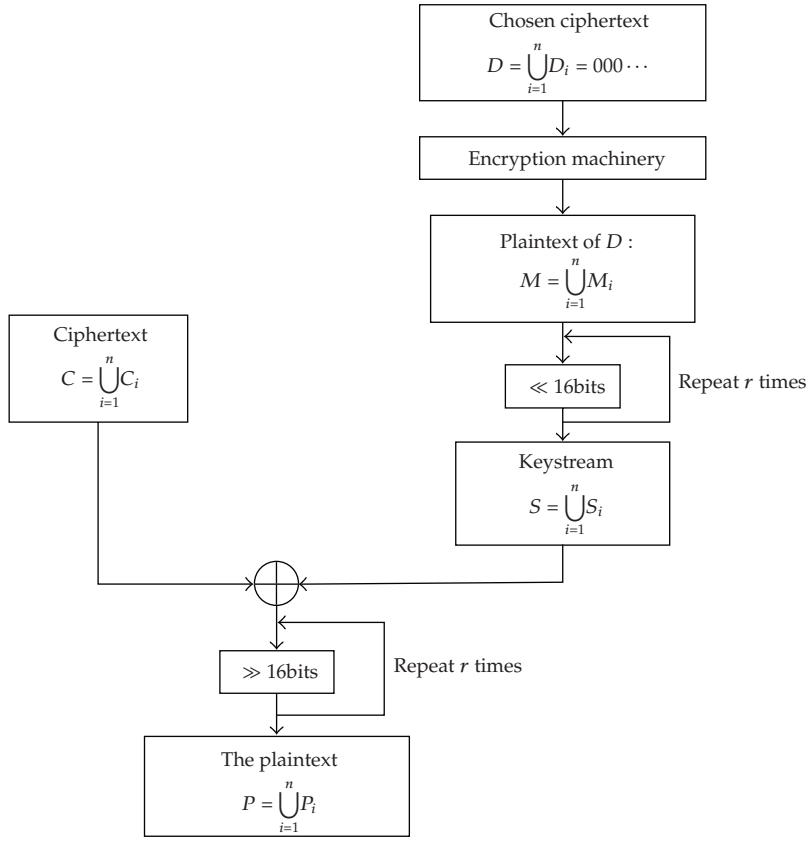
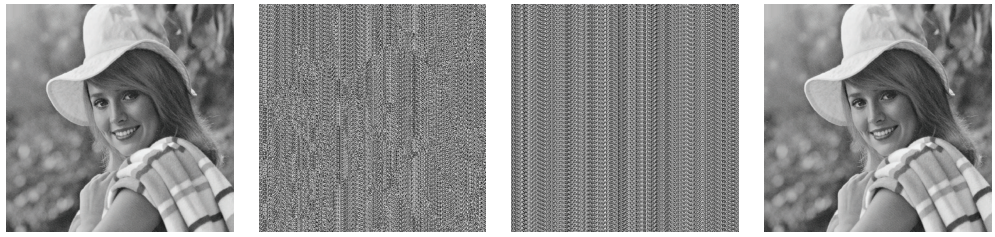


Figure 3: Flowchart of chosen ciphertext attacks.



(a) (b) (c) (d)

Figure 4: Results of the CCA: (a) the original image P; (b) ciphered image C of P; (c) plain image of the ciphered image $D = 0000$; (d) recovered image.

The flowchart of this attack is given in Figure 5 and Figure 6 shows a recovered Lena image from the ciphertext by the known plaintext attack using a known pair of plaintext/ciphertext of Jet.

Therefore, we can draw the following conclusion from the above three attacks.

Theorem 4.1. *The Wang-Yu block cipher is not secure (i.e., not one-way) under any of the following attacks: the chosen plaintext attacks, the chosen ciphertext attacks and the known plaintext attacks.*

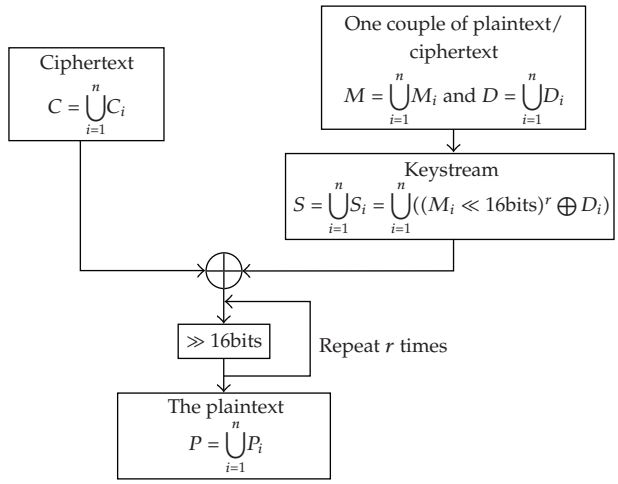


Figure 5: Flowchart of known plaintext attacks.

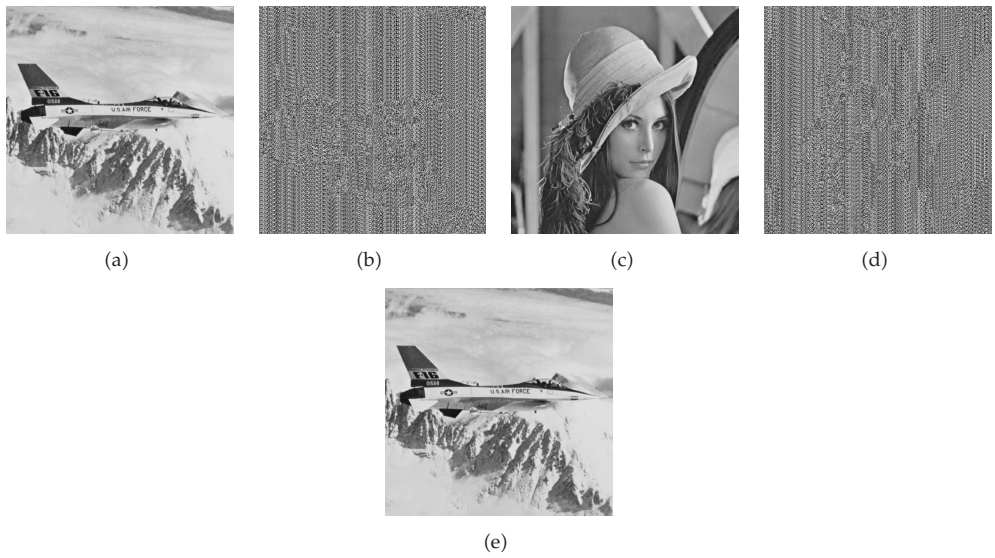


Figure 6: Results of the KPA: (a) the original image P; (b) ciphered image C of P; (c) plain image M; (d) ciphered image D of M; (e) recovered image from C.

5. Improvement

The ciphertext is independent from the keystream, which makes the encryption algorithm presented in [12] vulnerable to the above attacks. To enable the block cipher to be secure against the above attacks, a reasonable solution is shown. Actually, the ciphertext is sensitive to the change of the keystream and the plaintext in our improvement, which results in a cipher with enhanced security. In this section, we present the modification as follows.

(i) Encryption of $P_i \in \{0,1\}^{64}$: $C_i = E_k(P_i)$.

(1) Initialization: $B_i^{(0)} = P_i$; $l = 1$; $r \in_R \{0,1\}^*$; CMS table$\leftarrow k$; $P_0 = 1$; $C_0 = 1$.
Here, $k \in_R \{0,1\}^{64}$ is the secret key of the block cipher.

(2) $c(l) \in_R [0,3]$,
$d = (P_{i-1} \oplus C_{i-1}) \mod 128$,
$x_0 \leftarrow$ value of cth column, dth row in the CMS table.

(3) For $j = 1$ to 64:

(a) if $c(l) = 0$: $x_j = \mu x_{j-1}(1 - x_{j-1})$;
(b) if $c(l) = 1$: $x_j = \mu \sin(\pi x_{j-1})$;
(c) if $c(l) = 2$: $x_j = \mu \cos(\pi |x_{j-1} - 0.5|)$;
(d) if $c(l) = 3$: $x_j = 1 - \mu |x_{j-1} - 0.5|$.

(4) For $j = 1$ to 64:

$$s_j = \begin{cases} 1, & x_j \geqslant \overline{x}, \\ 0, & x_j < \overline{x}, \end{cases} \tag{5.1}$$

where $\overline{x} = 0.5$. The keystream is $SK_i^{c(l)} = s_1 s_2 \cdots s_{64}$.

(5) Encryption transformation:

$$\begin{aligned} B_i^{(2l-1)} &= B_i^{(2l-2)} \oplus SK_i^{c(l)}, \\ B_i^{(2l)} &= B_i^{(2l-1)} \ll 16 \text{ bits}. \end{aligned} \tag{5.2}$$

(6) If $l = r$, goto step (7);
else $l \leftarrow l + 1$; $d \leftarrow d + 1$; goto step (2).

(7) $C_i = B_i^{(2l)}$.

(ii) Decryption of $C_i \in \{0,1\}^{64}$: $P_i = D_k(C_i)$

Parameter and keystream generations here are the same as those in the encryption. The only difference is that the equations in step (5) should be replaced by

$$\begin{aligned} B_i^{(2l-1)} &= B_i^{(2l)} \gg 16 \text{ bits}, \\ B_i^{(2l-2)} &= B_i^{(2l-1)} \oplus SK_i^{c(l)}. \end{aligned} \tag{5.3}$$

6. Security and Efficiency of the Improved Scheme

The block cipher algorithm in [12] is not secure since its keystream is reused each time, which makes the keystream easy to recover. Our proposed version of the scheme is designed in a PCBC (Propagating Cipher Block Chaining) mode [13]. Through this mode, an attacker cannot discover the relationship among the keystream, the plaintext and the ciphertext. In order to show that our improved cipher is secure, we consider the three

kinds of attacks described previously to recover the keystream in our scheme. As a result, we find that the chosen plaintext attacks, the chosen ciphertext attacks and the known plaintext attacks all fail if the encryption algorithm and the decryption algorithm follow our proposal.

Moreover, the described attacks are harmless for the enhanced scheme. We assume that SK_i is the keystream used in the encryption of ith block. The keystream SK_i is computed as follows:

$$SK_i = \mathrm{H}_k(P_{i-1}, C_{i-1}), \quad \text{for } i = 1, 2, \ldots, L. \tag{6.1}$$

Here, P_{i-1} and C_{i-1} are the plaintext block and the ciphertext block with index $i - 1$, respectively; $\mathrm{H}_k()$ is a one-way function from chaotic map iterations. Obviously, even if an attacker knows P_{i-1} and C_{i-1}, he cannot get SK_i without the secret key.

Meanwhile, the above attacks cannot break our scheme since breaking the cipher is equivalent to knowing SK_i. However, knowing SK_i is impossible in our proposal. Thus, the algorithm is secure against the chosen plaintext attacks, the chosen ciphertext attacks and the known plaintext attacks.

In our scheme, one XOR and one MOD operations are added in the encryption of a plaintext block. The overload of the improved scheme does not influence the efficiency, compared with the Wang-Yu scheme. But our improved scheme achieves a high level of security.

The simulation for the proposed scheme is implemented in Matlab 7.0. Performance is measured on a 2.0 GHz Pentium Dual-Core with 1 GB RAM running Windows XP. The simulation results show that the average running speed of the Wang-Yu cipher and that of our improved cipher are 20.46 MB/s and 19.54 MB/s, respectively.

7. Conclusions

In this paper, three kinds of attacks are presented to break a recently proposed block cipher based on multiple chaotic systems. We show that the reuse of the keystream during the encryption iteration makes the Wang-Yu scheme insecure against the chosen plaintext attacks, the chosen ciphertext attacks and the known plaintext attacks. To enhance the security, we introduce a new method by updating the keystream in a way sensitive to the plaintext and the ciphertext.

Chaotic system is distinguished by its ergodicity and sensitivity to initial conditions and system parameters. These attributes allow the chaotic time series to be a promising alternative to the conventional cryptographic algorithms and image processing [14–21]. Fractal time series is distributed in a more random pattern than chaos time series does, due to the nondeterministic characteristic. Fractal time series differs from the conventional time series in the statistic properties [22–24]. Many open problems exist in this research area such as stationarity test problem [25], power spectrum problem [26, 27] and bound problem [28]. We are looking for possible ways to apply fractal time series to stochastic number simulation in cryptographic research.

Acknowledgment

This work has been supported by the Major Research plan of the National Natural Science Foundation of China (Grant no. 90718041), the National Natural Science Foundation of China (Grant nos. 10771072, 60703004 and 60873217), and the Research Fund for the Doctoral Program of Higher Education of China (Grant no. 20070269005).

References

[1] R. Matthews, "On the derivation of a "chaotic" encryption algorithm," *Cryptologia*, vol. 13, no. 1, pp. 29–42, 1989.

[2] M. Götz, K. Kelber, and W. Schwarz, "Discrete-time chaotic encryption systems. I. Statistical design approach," *IEEE Transactions on Circuits and Systems I*, vol. 44, no. 10, pp. 963–970, 1997.

[3] E. Alvarez, A. Fernández, P. Garcí, J. Jiménez, and A. Marcano, "New approach to chaotic encryption," *Physics Letters A*, vol. 263, no. 4–6, pp. 373–375, 1999.

[4] E. Biham, "Cryptanalysis of the chaotic-map cryptosystem suggested," in *Proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques (EUROCRYPT '91)*, vol. 547 of *Lecture Notes in Computer Science*, pp. 532–534, 1991.

[5] T. Stojanovski and L. Kocarev, "Chaos-based random number generators. I. Analysis," *IEEE Transactions on Circuits and Systems I*, vol. 48, no. 3, pp. 281–288, 2001.

[6] T. Stojanovski, J. Pihl, and L. Kocarev, "Chaos-based random number generators. II. Practical realization," *IEEE Transactions on Circuits and Systems I*, vol. 48, no. 3, pp. 382–385, 2001.

[7] F. Dachselt, K. Kelber, and W. Schwarz, "Discrete-time chaotic encryption systems—part III: cryptographical analysis," *IEEE Transactions on Circuits and Systems I*, vol. 45, no. 9, pp. 983–988, 1998.

[8] S. Lian, J. Sun, J. Wang, and Z. Wang, "A chaotic stream cipher and the usage in video protection," *Chaos, Solitons and Fractals*, vol. 34, no. 3, pp. 851–859, 2007.

[9] D. R. Frey, "Chaotic digital encoding: an approach to secure communication," *IEEE Transactions on Circuits and Systems II*, vol. 40, no. 10, pp. 660–666, 1993.

[10] N. K. Pareek, V. Patidar, and K. K. Sud, "Cryptography using multiple one-dimensional chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 10, no. 7, pp. 715–723, 2005.

[11] N. K. Pareek, V. Patidar, and K. K. Sud, "Discrete chaotic cryptography using external key," *Physics Letters A*, vol. 309, no. 1-2, pp. 75–82, 2003.

[12] X. Wang and Q. Yu, "A block encryption algorithm based on dynamic sequences of multiple chaotic systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 2, pp. 574–581, 2009.

[13] J. Kohl, "The use of encryption in kerberos for network authentication," in *Advances in Cryptology*, vol. 435 of *Lecture Notes in Computer Science*, pp. 35–43, 1990.

[14] Z. H. Liu, "Chaotic time series analysis," *Mathematical Problems in Engineering*, vol. 2010, Article ID 720190, 31 pages, 2010.

[15] E. G. Bakhoum and C. Toma, "Dynamical aspects of macroscopic and quantum transitions due to coherence function and time series events," *Mathematical Problems in Engineering*, vol. 2010, Article ID 428903, 13 pages, 2010.

[16] C. Cattani and A. Kudreyko, "Application of periodized harmonic wavelets towards solution of eigenvalue problems for integral equations," *Mathematical Problems in Engineering*, vol. 2010, Article ID 570136, 8 pages, 2010.

[17] G. Mattioli, M. Scalia, and C. Cattani, "Analysis of large amplitude pulses in short time intervals: application to neuron interactions," *Mathematical Problems in Engineering*, vol. 2010, Article ID 895785, 14 pages, 2010.

[18] S. Y. Chen, Y. F. Li, and J. Zhang, "Vision processing for realtime 3-D data acquisition based on coded structured light," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 167–176, 2008.

[19] S. Y. Chen, Y. F. Li, Q. Guan, and G. Xiao, "Real-time three-dimensional surface measurement by color encoded light projection," *Applied Physics Letters*, vol. 89, no. 11, Article ID 111108, 2006.

[20] S. Y. Chen and Y. F. Li, "Vision sensor planning for 3-D model acquisition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 5, pp. 894–904, 2005.

[21] S. Y. Chen, Y. F. Li, J. Zhang, and W. Wang, *Active Sensor Planning for Multiview Vision Tasks*, Springer, Berlin, Germany, 2008.

[22] M. Li, "Fractal time series—a tutorial review," *Mathematical Problems in Engineering*, vol. 2010, Article ID 157264, 26 pages, 2010.
[23] M. Li and J. Y. Li, "On the predictability of long-range dependent series," *Mathematical Problems in Engineering*, vol. 2010, Article ID 397454, 9 pages, 2010.
[24] M. Li, "Generation of teletraffic of generalized Cauchy type," *Physica Scripta*, vol. 81, no. 2, 10 pages, 2010.
[25] M. Li, W. S. Chen, and L. Han, "Correlation matching method of the weak stationarity test of LRD traffic," *Telecommunication Systems*, vol. 43, no. 3-4, pp. 181–195, 2010.
[26] M. Li and S. C. Lim, "Power spectrum of generalized Cauchy process," *Telecommunication Systems*, vol. 43, no. 3-4, pp. 219–222, 2010.
[27] M. Li and S. C. Lim, "A rigorous derivation of power spectrum of fractional Gaussian noise," *Fluctuation and Noise Letters*, vol. 6, no. 4, pp. C33–C36, 2006.
[28] M. Li and W. Zhao, "Representation of a stochastic traffic bound," *Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1368–1372, 2010.