Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2011, Article ID 938454, 12 pages
doi:10.1155/2011/938454

Research Article

Enhanced Cryptography by Multiple Chaotic Dynamics

Jianyong Chen,[1] Junwei Zhou,[1] Kwok-Wo Wong,[2] and Zhen Ji[1]

[1] *Shenzhen City Key Laboratory of Embedded System Design, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China*
[2] *Department of Electronic Engineering, City University of Hong Kong, Hong Kong*

Correspondence should be addressed to Jianyong Chen, cjyok2000@hotmail.com

Received 1 December 2010; Accepted 22 December 2010

Academic Editor: Ming Li

A potential security vulnerability of embedding compression in a chaos-based cryptography is studied. Furthermore, a scheme for improving its security is proposed. This correspondence considers the use of multiple chaotic dynamics and drive chaotic trajectory by both plaintext sequence and initial values of a chaotic map. Chaotic trajectory is used for encryption that is never reused for different plaintext. This makes that scheme naturally resist chosen plaintext attack and cipher text-only attack. Its strong security is justified by the key space, key sensitivity, and tests of random number sequences. The results show that the security of the proposed scheme is stronger than the latest algorithm especially in resisting chosen plaintext attack, while its performance is not sacrificed.

1. Introduction

Data compression and encryption become more and more important in multimedia communication. In order to improve both performance and security of multimedia application, it is worthwhile to joint compression and encryption in a united process [1–4]. In comparison with the classical separate compression-encryption schemes, the united scheme is more secure and effective [5]. Meanwhile, in the AES system, the zero padding can degrade coding efficiency, due to the block nature of AES. Another classical approach to provide simultaneous compression and encryption is to join a traditional entropy coder in a stream cipher, for example, RC4. Unfortunately, inappropriate integration with an initialization vector of RC4 leads to severe security vulnerability [5]. In order to improve the performance, two distinct research directions are studied recently. One is embedding key-based confusion and diffusion characteristics in existing compression algorithms such as entropy coding

[3, 4]. The scheme in [3] is based on multiple Huffman tables and achieves encryption and compression simultaneously via swapping the left and right branches of the Huffman tree. Moreover, some approaches are proposed to embed cryptography in arithmetic coding with key-based interval split [4].

Another approach is to incorporate compression in cryptography [2]. It is reported that the compression property can be embedded in chaos-based cryptography, so that the total time consumption of this scheme is less than the separation of encryption and compression. This scheme mixes block cipher and stream cipher processes. In the block cipher process, more probable symbols are encrypted by searching location of the plaintext symbol in the dynamic lookup table. While in the stream cipher process, less probable ones are masked by a pseudorandom bitstream. The block cipher could be considered as a variant of Baptista-type cryptosystem [6]. The Baptista-type cryptosystem is a cipher searching plaintext symbols based on a chaotic pseudorandom sequence. With its secret key, the original search trajectory is regenerated in decryption and the correct plaintext symbols are retrieved. It is found that the Baptista-type cryptosystem and some of its variants are vulnerable to various attacks, especially to chosen-plaintext attacks [7, 8]. In order to counter the attacks, remedial operations were suggested [9, 10]. Unfortunately, after studying the remedial method, it is found that the remedial method is inconvenient to improve security of [2]. Firstly, the remedial method is vulnerable to resist brute-force attack for it cuts down key space. Moreover, its speed is slow because of complicated operation. Therefore, it is necessary to enhance security of this class of chaos-based cryptosystems. For the mask model, there exists a potential safety risk of recovering pseudorandom number sequences and need to be solved.

In proposed scheme, we address this problem by disturbing the search of chaotic trajectory by an additional chaotic dynamics where confusion property is made more complicated while compression radio is not degraded evidently. Two chaotic maps are required in proposed cryptosystem: one map encrypts the first plaintext block. Then the other map encrypts next plaintext using final output of the first map as initial value. Short-period effect is also analyzed in details in terms of the drawback of chaotic dynamics in finite precision systems. Security analysis shows that the enhanced scheme can effectively repair the potential vulnerability and strengthen the security. Simulation results show that security of our proposed scheme is evidently improved and its compression radio is not degraded. Moreover, the key space of the multiple chaotic dynamics approach is enlarged and thus could effectively resist brute-force search attack.

The rest of this paper is organized as follows. In Section 2, cryptanalysis of embedding compression in chaos-based cryptography is presented. Its enhanced scheme could be found in Section 3. Analyses and simulation results can be given in Sections 4 and 5, respectively. In Section 6, conclusions will be drawn.

2. Cryptanalysis of Embedding Compression in Chaos-Based Cryptography

In recent years, there is an increasing trend of designing ciphers based on chaos [11–14]. This is because chaotic systems are sensitive to the initial condition and the system parameters [15]. These properties are desirable in cryptography. Moreover, the knowledge on chaos and nonlinear dynamics can be applied in the field of cryptography. Embedding compression in chaos-based cryptography proposed in [2] could be considered as hybrid cipher with both block cipher and stream cipher, which could be seamlessly embedded in image and

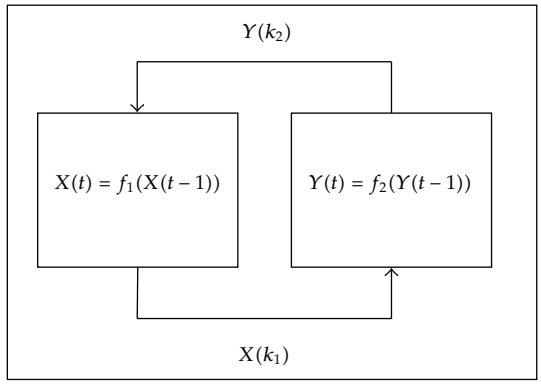

Figure 1: Structure of the proposed scheme.

multimedia systems [3, 16, 17]. Firstly, plaintext blocks with high probability are firstly encrypted by searching in the lookup table named block cipher, as is also named search model. After that, all the cipher text as well as plaintext with low probability is masked by a pseudorandom bitstream with stream ciphers, as is also named mask model.

Here, vulnerability of mask model is presented. The random number sequences used in mask model are generated by chaotic trajectory, and it is simply given as (2.1). The initial value x_0 and control coefficient b could be considered as pseudorandom number seed, and pseudorandom number sequences m are real key of this model. In a chosen plaintext attack, the attacker requests the ciphertext of $P_1 = \{s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1 \cdots\}$. The lookup table only has one symbol s_1, the needed iteration of block cipher is one time. Thus, the output of block cipher is composed of number 1. Now, the attacker knows output of block cipher and ciphertext exactly, and then he can obtain mask sequences without any knowledge of pseudorandom number seed. Even though the scheme proposed in [2] is hybrid cryptosystem, after the mask sequences have been obtained, the attacker could recover part of ciphertext. Furthermore, the cryptosystem becomes a single operation of block cipher which has been found vulnerable [7]

$$m = g_1(x_0, b). \tag{2.1}$$

3. The Proposed Scheme

3.1. Basic Principle

In the proposed scheme, the time series used in cryptosystem is generated by two chaotic systems, as shown in Figure 1. Moreover, the time series also could be fractal time series which is random as reported in [18–20]. Reference [21] discusses the random data generation of the type discussed in [22]. For sake of simplicity, only chaotic-based time series is studied in proposed paper. The first one is an arbitrary chaotic map stated in (3.1), the second one, as given by (3.2), is also a chaotic map and its chaotic trajectory is different from that of the first ones. There are two constraint conditions. Firstly, if the same type chaotic maps are adopted,

the control coefficients must keep unequal. Secondly, the phase space of chaotic map must be in the same range

$$X_t = f_1(X_{t-1}), \tag{3.1}$$

$$Y_t = f_2(Y_{t-1}). \tag{3.2}$$

Here, the searching model, that is, the first process of proposed scheme, is presented. The initial conditions are arbitrarily assigned with X_0. First of all, the sender performs k_1 iterations on (3.1) until chaotic state X_{k_1} locates in the phase space mapped to the first plaintext block. After the first symbol was encrypted, X_{k_1} is selected as the initial state of the second chaotic map and used for encrypting the second plaintext block. The sender performs k_2 iterations on (3.2) and Y_{k_2} locates in the phase space of the second plaintext block. And then Y_{k_2} is selected as initial state of (3.1) in the second round. The third plaintext block is encrypted by (3.1). These operations repeat until all the plaintext blocks have been encrypted. The cipher text is a collection of the number of iterations $\{k_1\ k_2\ k_3 \cdots k_m\}$ obtained in each round. Therefore, there are new initial states related to previous plaintext block for encrypting each plaintext block.

3.2. Chaotic Maps for the Proposed Scheme

To illustrate the proposed scheme, logistic map is used as chaotic map which is a typical chaotic system widely used in cryptosystem [1, 2, 6], as shown in (3.3) and (3.4). The control coefficient b_0 and b_1 are unequal. Both of them are in the range of [3.9, 4] so as to avoid the nonchaotic regions according to [23]

$$x_t = b_0 x_{t-1}(1 - x_{t-1}), \tag{3.3}$$

$$y_t = b_1 y_{t-1}(1 - y_{t-1}). \tag{3.4}$$

To examine the properties of the multiple chaotic dynamics, the largest Lyapunov exponent of the discrete time series is computed by the method described in [24]. The parameters of chaotic map b_0 and b_1 are arbitrarily selected from the range of [3.6, 4]. The iteration times of two chaotic maps are randomly selected. The largest Lyapunov exponent is tested thousands of times, all of test results are bigger than 0. It means that the system is chaotic at various situations that assure the chaotic status at encryption and decryption processes.

3.3. Encryption Procedures

Similar to [2], the encryption procedure is hybrid one. More probable symbols are encrypted by searching in the lookup table, while less probable ones are masked by a pseudorandom bitstream as performed in stream ciphers. In the searching model, the phase space of the logistic map is divided into a number of equal-width partitions. Each partition associates with a possible plaintext symbol. The greater the probability of occurrence of the symbol

is, the more phase-space partitions it maps to. A secret chaotic trajectory produced by the multiple chaotic dynamics system is used to search the partition mapped to the plaintext symbol. The number of iterations of the logistic map for searching each plaintext symbol is the length of the searching trajectory, which is then taken as the cipher text [6]. Meanwhile, the pseudorandom sequence is generated from the secret chaotic trajectory, and it will be used in mask model. After searching model has been processed, Huffman code is performed for all the collected number of iterations. At the last step, the intermediate sequence and the less probable plaintext sequence are masked by the binary mask sequence which is generated in searching model.

Without loss of generality, the plaintext is assumed as a sequence of symbols. The encryption procedures are as follows.

Step 1. Scan the whole plaintext sequence to find out the number of occurrence for each plaintext symbol. Then, select the top symbols, and construct map table according to the probability.

Step 2. Encrypt each more probable plaintext symbol sequentially according to the method in [2]. The different is that the two chaotic maps are used rotationally. For instance, (3.3) is used for encrypting the jth plaintext block, the end state of (3.3) is adopted as the initial state of (3.4). Then $(j+1)$th block is encrypted by (3.4). In the each iteration, eight masking bits are extracted from the least significant byte of the chaotic trajectory. They are appended to form a binary mask sequence for later use in mask model.

Step 3. After all the plaintext blocks have been processed, a Huffman tree is built for all the collected number of iterations, including zero. When the Huffman tree is built, the number of iterations and the special symbol are replaced by the corresponding variable-length Huffman code to form the intermediate sequence r.

Step 4. The intermediate sequence r should be masked by the binary mask sequence, which is a stream cipher named mask model. This mask model is different from the original method proposed in [2] as shown in (3.5). The ciphertext of block j is given by (3.5) where r_j, m_{j-1} and C_j are intermediate sequences, mask sequences and ciphertext sequence packaged in 32-bit block, respectively. The m_{-1} is initial value pseudorandom number sequence that could be considered as cipher key of proposed scheme

$$c_j = \left(m_{j-1} + r_j\right) \bmod 2^{32}. \tag{3.5}$$

3.4. Decryption Procedures

Before the decryption, initial values of the chaotic map, which are the secret key, must be delivered to the receiver secretly. The key includes the control parameters b_0, b_1, m_{-1} and the initial values x_0 of the chaotic maps. Moreover, the plaintext probability information must also be available to the receiver for reconstructing the symbol mapping table. With this information, the receiver is able to reproduce the secret chaotic trajectory used in encryption. The decryption process is similar to the encryption one. The receiver regenerates the chaotic sequence from the secret key, and then looks up the plaintext symbol from the table. The decryption processes for decrypting jth are as follows.

Step 1. Unmask the intermediate sequence r_j according to m_{j-1} of (3.5).

Step 2. According to intermediate sequence r_j, iterate the chaotic map using the shared secret parameters and the initial conditions to regenerate the chaotic trajectory as used in encryption. Decode the variable-length Huffman code using the shared Huffman tree to find out the number of iterations required. If the number is zero, this means that the block was only encrypted in mask mode and the block is copied directly as the output. Otherwise, iterate the chaotic map with the nonzero number of iterations and determine the final partition visited by the chaotic trajectory. Then, extract the mask bits from the binary representation of each chaotic map output to form the binary mask sequence m_j. Go to Step 1 and unmask the intermediate sequence r_{j+1}.

Step 3. Perform Steps 1 and 2 repeatedly until whole ciphertext blocks are decrypted.

4. Security Analysis

4.1. One-Time Pad Attacks

The security of the original Baptista-type cryptosystem is analyzed in [7, 8] and some effective attacks were suggested there. The major problem of this cryptosystem is that the search trajectory is solely determined by the secret key. The same trajectory is used for all plaintext sequences unless the secret key is changed. As a result, the cryptanalyst can easily launch a chosen-plaintext attack to recover the whole chaotic trajectory. Details of the attack can be found in [7, 8]. Therefore, some researchers insist that Baptista-type cryptosystem can not meet even the most basic security requirements. The fatal vulnerability of Baptista's scheme is one-time pad attack which is a considered as a kind of chosen plaintext attack. The plaintext block of each state of chaotic trajectory is obtained by way of one-time pad attacks. As shown in Figure 2, the chaotic trajectory U is generated from key based chaotic map, and the corresponding plaintext sequence V is obtained by one-time pad attacks [7]. Since U remains unchanged for different plaintext sequence, if one-time pad requests the cipher text of the following plain text $P_1 = (s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1 \cdots)$, the corresponding cipher text is $C_1 = (5\ 3\ 2\ 2\ 2\ 3\ 2\ 3\ 2\ 2\ 3\ 2 \cdots)$. Examining the cipher text, the attacker knows that the value $\{v_5, v_8, v_{10} \cdots\}$ is s_1. Then the attacker could get partial information about V. To complete knowledge of the symbolic sequence V, the attacker requests the cipher text of all kinds of plaintext sequence similar to P_1. After obtaining enough sequence V, attacker does not need to know initial value of chaotic map, while he can know exactly the plaintext corresponding to ciphertext.

From the above analysis, it is known that the one-time pad attacker is to obtain corresponding plaintext block sequences V. The variables $V_1 \cdots V_2$ denote the partial information of corresponding plaintext block sequences V. The way is to request the cipher text of the plaintext like P_1, and then obtain partial information V_1 of V. After the attacker has requested the cipher text of all 256 kinds of plaintext, full knowledge of the symbolic sequence V is gained as shown in (4.1) where e is plaintext bit length. The value of e is set 8 because symbol in ASCII is 8-bit. In proposed scheme, the encryption procedure not only depends on the initial value, but also relates to the plaintext. Each distinct plaintext block leads to a different search trajectory for encryption. The knowledge of a search trajectory, for a particular plaintext block, is useless for other plaintext block. Moreover, the chaotic maps are iterated sequentially, the confusion of encryption is diffused. These lead to the difficulty

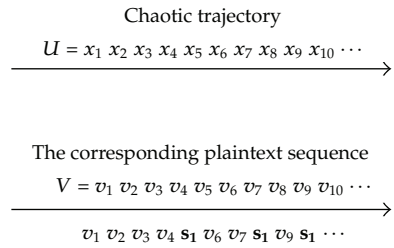

Figure 2: Chaotic trajectory and the corresponding plaintext sequence.

in attacking the proposed cryptosystem using known/chosen-plaintext attack. In Section 4.2, we know that the trajectory is different for different plaintext $\{P_1\ P_2\ P_3\cdots\}$. Therefore, (4.1) is unavailable in proposed scheme, and the attacker couldnot get the full knowledge of V

$$V = V_1 \cup V_2 \cdots V_{2^e-1} \cup V_{2^e}. \tag{4.1}$$

4.2. The Security of Block Model and Mask Model

The block cipher of Embedding Compression in Chaos-Based Cryptography could be considered as a variant of Baptista-type cryptosystem [6]. Here, the reasons of vulnerability were reviewed. Considering the ciphertext sequence $C_1 = (1\ 2\ 2\ 1\ 3\ 1\cdots)$ and $C_2 = (2\ 1\ 3\ 2\ 1\ 1\cdots)$, the corresponding trajectories are U_1 and U_2 as shown in Table 1. For comparison, the original scheme Baptista-type cryptosystem [6] is present. It is found that the chaotic trajectory U_1 is distinct from U_2 in proposed scheme. However, they keep the same in Baptista's type cryptosystem.

The scheme presented in [25] is a complex combination of multiple chaotic maps. In proposed scheme, the interaction among chaotic trajectories is existent. After encrypting current plaintext block, the last state of chaotic map is selected as the initial state of next chaotic trajectory to encrypt next plaintext block. Therefore, the initial value used for encrypting current plaintext block is related to last ciphertext, and thus different plaintext sequences lead to distinct initial values for encryption. It is useless to launch a chosen plaintext attack by regenerating the chaotic trajectory, since each different plaintext is encrypted by unique chaotic trajectory. Meanwhile, the additional chaotic dynamics results in a more secure cryptosystem for it increases the confusion in the encryption process and expands key space.

After searching model has been performed, pseudorandom bitstream is generated by chaotic trajectory, the less probable symbols are encrypted by masking them with the pseudorandom bitstream. In proposed scheme, the pseudorandom number sequences are also generated by chaotic trajectory which is related to plaintext sequences. It is simply given as (4.2) where P is plaintext sequences. In a chosen plaintext attack, the attacker requests the ciphertext of $P_1 = \{s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\ s_1\cdots\}$. In the same way presented in Section 2, it can obtain mask sequences without any awareness of pseudorandom number seed. However, the pseudorandom number sequences only relate to the plaintext P_1. For other plaintext sequences, the mask sequences m are totally different. Hence this kind of

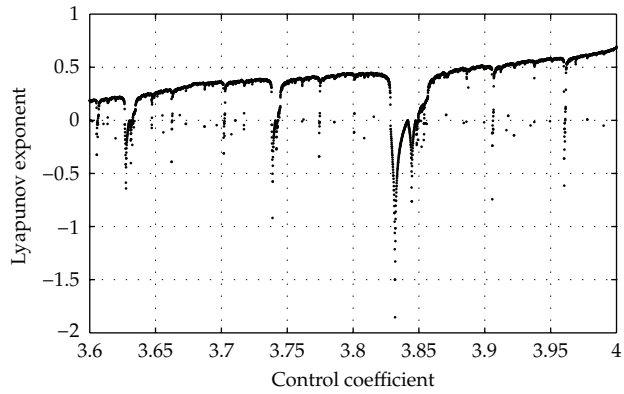

Figure 3: A plot of the Lyapunov exponent computed at increments of 0.0001 where control coefficient b is in the range of [3.6, 4].

chosen plaintext attack doesnot work in proposed method. Now, both block cipher and stream cipher of [2] are enhanced

$$m = g_2(x_0, b, P). \tag{4.2}$$

4.3. Periodicity of Multiple Chaotic Dynamics

The period of outputs in chaotic region can be looked on as infinite. However the actual period is limited by precision format of digital computer. Short period of chaotic trajectories generated by finite precision system [26] has been an obstacle in employing chaotic dynamics for cryptographic purpose. Short period makes the chaotic trajectories reused frequently in encryption process. This is equivalent to reuse key and weaken the security of the cryptosystem. The control parameter b_0 and b_1 should be chosen carefully to avoid nonchaotic regions [2, 23] whose Lyapunov exponent is negative. In proposed scheme, Lyapunov exponent is chosen as an indicator. It defines the region where the system behaves chaos [27] as shown in Figure 3. We calculated the period of logistic map when the control coefficient b_0 and b_1 is in the region where Lyapunov exponent is positive. Results show that all the length of period in chaotic region is far larger than 10^7 in double precision system which is a reasonable period length in practical cipher applications [26].

For plaintext perturbation and interaction of multiple chaotic maps, the period of multiple chaotic maps tends to infinity in theory. Even in the worst case, the period of multiple chaotic maps is the same as that of single map.

5. Simulation Results

To implement the proposed cryptosystem, the control parameters are arbitrarily selected as $b_0 = 3.999999991$ and $b_1 = 3.999991$. The initial condition x_0 is arbitrarily set to 0.3388. The plaintext block is encrypted in bytes and the phase space of chaotic map is divided into 256 equal-width partitions. The maximum number of iterations for the search mode is 15. The

Table 1: Chaotic trajectories for various ciphertext sequences.

Method	Ciphertext	
	Proposed scheme	Baptista's type
$C_1 = (1\ 2\ 2\ 1\ 3\ 1\cdots)$	$U_1 = x_1y_1y_2x_1'x_2'y_1'x_1''x_2''x_3''y_1''\cdots$	$U_1 = x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}\cdots$
$C_2 = (2\ 1\ 3\ 2\ 1\ 1\cdots)$	$U_2 = x_1x_2y_1x_1'x_2'x_3'y_1'y_2'x_1''y_1''\cdots$	$U_2 = x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}\cdots$

Table 2: Ciphertext-to-plaintext ratio of Calgary corpus files.

File	Multiple chaotic equations	Scheme in [2]	Huffman coding
pic	32.77%	32.93%	20.92%
book1	83.36%	83.39%	57.08%
paper2	83.94%	84.23%	58.50%
geo	83.94%	84.20%	72.12%
paper3	84.30%	84.79%	59.56%
paper4	84.24%	85.90%	62.31%
book2	84.99%	85.09%	60.37%
progp	85.63%	86.09%	62.13%
progl	86.09%	86.31%	60.62%
paper5	86.16%	88.13%	66.12%
paper1	86.94%	87.31%	63.64%
paper6	87.02%	87.71%	64.31%
news	87.89%	88.03%	65.47%
progc	88.29%	89.31%	66.63%
obj1	88.50%	89.63%	80.68%
bib	89.00%	89.27%	65.78%
obj2	92.80%	92.95%	79.17%
trans	93.17%	93.50%	70.15%

proposed algorithm is implemented in C++ programming language running on a personal computer with an Intel Core(TM) 2 2.00 GHz processor and 2 GB memory.

5.1. Compression Ratio, Encryption and Decryption Speed

To test the compression capability of the proposed scheme, 18 distinct files of different types are used, including text, executable file, geophysical data and picture. They are standard files from the Calgary Corpus. These files are encrypted using the proposed scheme and the algorithm suggested in [2], respectively. Only 16 probable plaintext symbols are selected and they are all mapped to one table [2]. The simulation results are listed in Table 2. It shows that all files can be compressed effectively using proposed approach. The compression ratios of our scheme are same as those reported in [2].

The encryption and decryption speed of Baptista-type chaotic cryptosystem depends on the average number of iterations required. The encryption and decryption time on the Calgary Corpus files [28] can be found in Table 3. For comparison purpose, the encryption and decryption time using the algorithm in [2] are also listed. The results show that total processing time of proposed scheme is almost the same as that of the scheme proposed in [2].

Compared with existing chaotic cryptographic schemes, the proposed scheme does not take any additional computation. Tables 2 and 3 indicate that the multiple chaotic

Table 3: Encryption and decryption time of calgary corpus files.

File	Size (byte)	Encryption			Decryption		
		Proposed algorithm (sec)	Scheme in [2] (sec)	AES (128) + Huffman coding	Proposed algorithm (sec)	Scheme in [2] (sec)	AES (128) + Huffman coding
paper5	11,954	0.008437	0.008019	0.01707	0.010243	0.009611	0.01694
paper4	13,286	0.009042	0.009897	0.01750	0.010604	0.010042	0.01731
obj1	21,504	0.010703	0.010545	0.02020	0.013569	0.013434	0.01435
paper6	38,105	0.017122	0.017918	0.02451	0.020636	0.019127	0.02431
progc	39,611	0.017172	0.017602	0.02773	0.021448	0.019723	0.02590
paper3	46,526	0.009028	0.020962	0.02567	0.023847	0.021379	0.02605
progp	49,379	0.020326	0.022016	0.03061	0.024517	0.022455	0.02719
paper1	63,161	0.022271	0.023657	0.03139	0.026781	0.024091	0.02799
progl	71,646	0.028402	0.030249	0.03458	0.03388	0.030363	0.03147
paper2	82,199	0.032868	0.036339	0.03826	0.037825	0.033343	0.03387
trans	93,695	0.03291	0.034544	0.04968	0.044538	0.040471	0.03680
geo	102,400	0.036189	0.039732	0.04919	0.043722	0.041091	0.03169
bib	111,261	0.040427	0.04336	0.05506	0.050979	0.045187	0.04169
obj2	246,814	0.078032	0.084454	0.11073	0.106291	0.096167	0.06881
news	377,109	0.126956	0.130061	0.15337	0.159993	0.139761	0.09680
pic	513,216	0.176328	0.201815	0.11150	0.108684	0.12076	0.05698
book2	610,856	0.215156	0.220239	0.22673	0.24827	0.215095	0.14088
book1	768,771	0.279308	0.289482	0.28122	0.3082612	0.273162	0.16598

dynamics do not lead to a sacrifice of performance due to enhance of security. For the classical separate compression-encryption schemes, the test results of speed are directly chosen from [2]. Table 3 shows that speed of both proposed scheme and scheme in [2] is faster than that of the separate compression-encryption scheme.

5.2. Key Space and Key Sensitivity

To test the key sensitivity, the files from the Calgary Corpus are encrypted using different sets of secret key. Encryptions using proposed scheme were performed with only a small change of the parameters. The two resultant cipher text sequences are then compared bit-by-bit and the percentage of bit change is calculated. For parameters b_0, b_1, and x_0, the 15th digit after decimal point is changed by a minimal value. For m_{-1}, the least bits in different position are crossover from 0 to 1 or 1 to 0. The measured bit change percentages are 50.01%, 50.01%, 49.98% and 50.07% for b_0, b_1, x_0 and m_{-1}, respectively. The results show that bit change percentages are close to 50% and indicate that the cipher text is very sensitive to the key.

The freely-chosen key of the proposed scheme consists of the control parameters b_0, b_1 and the initial values x_0, together with the 32-bit initial cipher block value m_{-1}. The key sensitivity test indicates that the 15th bit is effective that the ciphertext is sensitive to that bit. Therefore, the control parameters b_0, b_1 and initial state x_0 are real numbers represented by double-precision format that are equivalent to 52 bits. According to Rule 5 of [23], b_0 and b_1 should avoid the nonchaotic regions. Here, similar to [2], the range of [3.9, 4.0] is adopted as

chaotic regions and this is equivalent to 46 bits. The total key space is then 52 + 46 + 46 + 32 = 176 bits, which is longer than 130 bits used in [2]. Similar to [12, 13], chaotic maps with a higher dimension can be chosen if a larger key space is required. As a result, the attacker has to meet with an even larger key space for resisting a brute-force search attack.

5.3. Randomness of the Binary Mask Sequence

The mask model is a stream cipher that masks plaintext symbols by a pseudorandom bitstream [2]. The randomness of the pseudorandom bitstream is also confirmed by the statistical test suite recommended by the U.S. National Institute of Standards and Technology (NIST) [29]. 300 sequences, each of bits, are extracted for testing, and they all pass the statistical tests including frequency, block frequency, cumulative sums, runs, longest run, rank, and fast Fourier transform (FFT). All of P-values are larger than 0.01. Therefore, the sequences are considered as random according to the NIST Special Publication 800–22 [29].

6. Conclusion

In order to strengthen the security of chaos-based cryptography with embedded compression, a general multiple chaotic dynamics is proposed to correlate the chaotic trajectory with the plaintext. As a result, both the two steps of embedding compression in chaos-based cryptography are enhanced. For the searching model, the security is enhanced to resist a chosen plaintext attack named one-time pad attack. In the mask model, a potential vulnerability is analyzed and repaired. Meanwhile, the key space of proposed scheme is enlarged to resist a brute-force search attack.

Acknowledgment

This work was supported by Shenzhen University Research and Development Fund (SZU R/D) with Grant no. 200903.

References

[1] S. E. Borujeni and M. Eshghi, "Chaotic image encryption design using Tompkins-Paige algorithm," *Mathematical Problems in Engineering*, vol. 2009, Article ID 762652, 22 pages, 2009.
[2] K. W. Wong and C. H. Yuen, "Embedding compression in chaos-based cryptography," *IEEE Transactions on Circuits and Systems II*, vol. 55, no. 11, pp. 1193–1197, 2008.
[3] C. P. Wu and C. C. J. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 828–839, 2005.
[4] H. Li and J. Zhang, "A secure and efficient entropy coding based on arithmetic coding," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 12, pp. 4304–4318, 2009.
[5] J. Zhou, O. C. Au, and P. H. W. Wong, "Adaptive chosen-ciphertext attack on secure arithmetic coding," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 1825–1838, 2009.
[6] M. S. Baptista, "Cryptography with chaos," *Physics Letters A*, vol. 240, no. 1-2, pp. 50–54, 1998.
[7] G. Álvarez, F. Montoya, M. Romera, and G. Pastor, "Cryptanalysis of an ergodic chaotic cipher," *Physics Letters A*, vol. 311, no. 2-3, pp. 172–179, 2003.
[8] G. Álvarez, F. Montoya, M. Romera, and G. Pastor, "Cryptanalysis of dynamic look-up table based chaotic cryptosystems," *Physics Letters A*, vol. 326, no. 3-4, pp. 211–218, 2004.
[9] J. Wei, X. Liao, K. W. Wong, T. Zhou, and Y. Deng, "Analysis and improvement for the performance of Baptista's cryptographic scheme," *Physics Letters A*, vol. 354, no. 1-2, pp. 101–109, 2006.

[10] S. Li, G. Chen, K. W. Wong, X. Mou, and Y. Cai, "Baptista-type chaotic cryptosystems: Problems and countermeasures," *Physics Letters A*, vol. 332, no. 5-6, pp. 368–375, 2004.
[11] I. Dalkiran and K. Danişman, "Artificial neural network based chaotic generator for cryptology," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 18, no. 2, pp. 225–240, 2010.
[12] N. Singh and A. Sinha, "Chaos based multiple image encryption using multiple canonical transforms," *Optics and Laser Technology*, vol. 42, no. 5, pp. 724–731, 2010.
[13] H. H. Nien, W. T. Huang, C. M. Hung et al., "Hybrid image encryption using multi-chaos-system," in *Proceedings of the 7th International Conference on Information, Communications and Signal Processing (ICICS '09)*, pp. 1–5, December 2009.
[14] J. He, H. Qian, Y. Zhou, and Z. Li, "Cryptanalysis and improvement of a block cipher based on multiple chaotic systems," *Mathematical Problems in Engineering*, vol. 2010, Article ID 590590, 14 pages, 2010.
[15] L. A. Aguirre and C. Letellier, "Modeling nonlinear dynamics and chaos: a review," *Mathematical Problems in Engineering*, vol. 2009, Article ID 238960, 35 pages, 2009.
[16] S. Y. Chen and Y. F. Li, "Determination of stripe edge blurring for depth sensing," *IEEE Sensors Journal*, vol. 11, no. 2, pp. 389–390, 2011.
[17] S. Y. Chen et al., "Improved generalized belief propagation for vision processing," *Mathematical Problems in Engineering*, vol. 2011, Article ID 416963, 12 pages, 2011.
[18] B. B. Mandelbrot, *Gaussian Self-Affinity and Fractals*, Springer, New York, NY, USA, 2002.
[19] C. Cattani, "Harmonic wavelet approximation of random, fractal and high frequency signals," *Telecommunication Systems*, vol. 43, no. 3-4, pp. 207–217, 2010.
[20] E. G. Bakhoum and C. Toma, "Mathematical transform of traveling-wave equations and phase aspects of quantum interaction," *Mathematical Problems in Engineering*, vol. 2010, Article ID 695208, 15 pages, 2010.
[21] M. Li, "Generation of teletraffic of generalized Cauchy type," *Physica Scripta*, vol. 81, no. 2, Article ID 025007, 2010.
[22] M. Li and W. Zhao, "Representation of a Stochastic Traffic Bound," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1368–1372, 2010.
[23] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 16, no. 8, pp. 2129–2151, 2006.
[24] M. T. Rosenstein, J. J. Collins, and C. J. De Luca, "A practical method for calculating largest Lyapunov exponents from small data sets," *Physica D*, vol. 65, no. 1-2, pp. 117–134, 1993.
[25] N. K. Pareek, Vinod 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.
[26] T. Addabbo, M. Alioto, A. Fort, A. Pasini, S. Rocchi, and V. Vignoli, "A class of maximum-period nonlinear congruential generators derived from the Rényi chaotic map," *IEEE Transactions on Circuits and Systems. I*, vol. 54, no. 4, pp. 816–828, 2007.
[27] C. M. Ou, "Design of block ciphers by simple chaotic functions," *IEEE Computational Intelligence Magazine*, vol. 3, no. 2, Article ID 4490261, pp. 54–59, 2008.
[28] ftp://ftp.cpsc.ucalgary.ca/pub/projects/text.compression.corpus.
[29] A. Rukhin et al., "A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications," National Institute of Standards and Technology (NIST), Gaithersburg, Md, USA, NIST Special Publication 800-22, April 2010, http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html.