A Comparative Study of Various Simultaneous Compression and Encryption Schemes Based On Chaotic Systems

Qiuzhen Lin†, Kwok-Wo Wong‡, Junwei Zhou‡, and Leo Yu Zhang‡

† College of Computer Science and Software Engineering, Shenzhen University
Nanhai Avenue 3688, Shenzhen 518060, P.R. China
‡ Department of Electronic Engineering, City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong SAR
Email: qiuzhlin@szu.edu.cn, itkwwong@cityu.edu.hk, junweizhou@msn.com, leocityu@gmail.com

Abstract – Recently, various schemes for performing compression and encryption simultaneously are designed by exploiting the dynamical properties of chaotic systems. They have shown promising performance in reducing data redundancy and maintaining information secrecy. In this paper, a comparative study of different chaos-based schemes for simultaneous compression and encryption will be presented. Experimental studies are conducted to show their strengths in compression performance. Moreover, security issues are analyzed.

1. Introduction

Recently, there is a growing interest in performing compression and encryption simultaneously [1, 2]. As the two operations are performed in a single step, a simpler system design and an improved operating efficiency can be achieved.

The first attempt for this joint scheme was realized by maintaining the secrecy of the statistical model used in arithmetic coding (AC) [3]. Some subsequent studies include multiple Huffman tree (MHT) [4], randomized AC (RAC) [5], interval-splitting AC (IAC) [6], secure AC (SAC) [1] and first-order RAC [7]. However, most of them are found insecure as MHT and IAC suffer from known-plaintext attack [8, 9] while RAC and SAC are vulnerable upon ciphertext-only attack [10, 11] and adaptive chosen-ciphertext attack [2], respectively.

In order to fix the above-mentioned security loopholes, chaotic systems are adopted for enhancement [12-16]. Making use of the high sensitivity to the initial conditions, chaotic systems are generally exploited in the design of stream ciphers, for masking the information bits. Besides this, it is found that source coding by iterating a piecewise linear chaotic map is equivalent to the optimal entropy coding, which is regarded as a generalized version of AC [17]. Thus, chaotic systems are actually a perfect alternative for performing simultaneous compression and encryption. Based on the capabilities provided by chaotic systems, chaos-based simultaneous compression and encryption schemes can be generally classified into three categories.

The first type considers chaotic systems as a pseudo-random number generator (PRNG), which keeps the secrecy of statistical model adopted in source coding. In [12], a chaos-based adaptive AC scheme is presented with the statistical model of AC controlled by a PRNG generated from a coupled chaotic system. As there is a design fault in [12], a modified version is suggested in [13]. A novel chaotic encryption scheme is designed by combining AC and logistic map [14], where the plaintext is combined with the pseudo-random bits generated from the logistic map to vary the statistical model of AC. Instead of the logistic map, an improved version [15] adopts a piecewise linear chaotic map to improve the coding efficiency and the distribution of iterated values. A new joint compression and encryption scheme based on Huffman coding and chaotic system is presented in [16], which determines the Huffman tree by the input message and a key stream generated from a chaotic map.

The second class of simultaneous compression and encryption schemes embeds compression into chaos-based encryption. A chaos-based cryptographic scheme with compression capability is suggested by partitioning the phase space according to the plaintext statistics [18]. A similar scheme is investigated in [19], which employs adaptive AC instead of Huffman coding to achieve a higher compression ratio. By dynamically updating the lookup table used for chaos-based encryption, a modified version of chaos-based joint compression and encryption is proposed to achieve a higher compression performance [20]. A novel chaos-based joint compression and encryption scheme is designed for generating variable-length ciphertext using a user-chosen parameter [21]. Instead of recording the total number of iterations of the chaotic map, it counts the number of distinct symbols visited by the chaotic search orbit and results in a better compression ratio.

The third kind of simultaneous compression and encryption schemes exploits both the compression and encryption capabilities offered by chaotic systems. A simultaneous arithmetic coding and encryption scheme using chaotic map is presented [22], where the model of the chaotic map is determined by a secret key controlled by another chaotic map. It leads to a higher security level than existing schemes based on traditional AC [1, 5, 6] as the number of secret models provided by the chaotic map is larger than that of the traditional AC. However, the

operating time can be very long due to the high-precision computation and the compression performance is affected by the insertion of separator symbols. Thus, an enhanced version is suggested [23], which encodes a block of variable number of symbols to a codeword within a small computational register. The operating efficiency is substantially improved and the compressed sequence is further processed by an additional diffusion operation. This modified version [23] strengthens the security of the original scheme [22] by having higher key and plaintext sensitivities.

In this paper, we perform a comparative study of various chaos-based schemes for simultaneous compression and encryption. Their strengths on compression performance are investigated. Security issues of these schemes are also discussed. To the best of our knowledge, this is the first work to compare the comprehensive performance of different chaos-based simultaneous compression and encryption schemes.

The rest of this paper is organized as follows. In Section II, three representative simultaneous compression and encryption schemes using chaotic systems are briefly reviewed. Experimental results are presented in Section III. Finally, conclusions are drawn in Section IV.

2. Some Representative Schemes

2.1. Chaos-based Adaptive AC Scheme

The scheme proposed in [12] adopts a stream cipher generated from a chaotic map to distort the statistical model of AC. Its model is illustrated in Fig. 1, where the seed for the chaotic map is considered as the secret key and is transmitted to the receiver through a secure channel. Let the input sequence be $S = s_1, s_2, ..., s_N$, among which there are n ($n \le N$) distinct alphabets as $A = \{a_1, a_2, ..., a_n\}$. The encoding procedures can be summarized as follows [13].

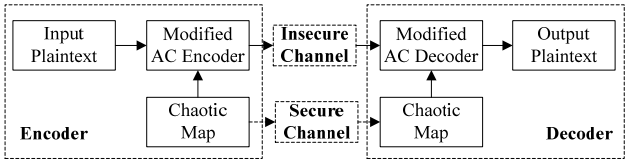

Fig. 1 Model of the chaos-based adaptive AC scheme [12].

Step 1: Initialize the symbol distribution according to a PRNG based on a coupled chaotic system. Set i=1.
Step 2: To encode a source symbol s_i, firstly sort the symbols in A according to their frequencies and denote the sorted sequence as $A' = \{a'_1, a'_2, ..., a'_n\}$ with $c'_j \ge c'_{j+1}$, where c'_j is the frequency count of a'_j ($1 \le j \le n$).
Step 3: Assume that s_{i-1} corresponds to $a'_{i'}$ in A' and then construct the near symbol array

$$T = \{a'_{i'-t}, a'_{i'-t+1}, ..., a'_{i'-1}, a'_{i'+1}, a'_{i'+t-1}, a'_{i'+t}\}$$

where t is a threshold to control the number of symbol frequencies used for reshuffling.
Step 4: Select the k-th element represented by a'_k in T, where k is a random number generated from a chaotic map.
Step 5: Interchange the positions of $a'_{i'}$ and a'_k in A', and then construct the statistical model according to A'. Use this secret statistical model to reduce the coding interval to that allocated to s_i.
Step 6: Increase the frequency count of s_i by one and reset the counters i=i+1. If $i \le N$, go to Step 2. Otherwise, output the final bit stream.

It should be noticed that when the first source symbol s_1 is encoded, the previous encoded symbol s_0 is not available. Thus, s_0 can be determined by a random number from the chaotic map. More details of this scheme can be found in [12, 13].

2.2. Baptista-based Joint Scheme

This scheme embeds compression capability into the Baptista-type chaotic cryptosystem [18]. The lookup table used in encryption is constructed by the probabilities of occurrence of plaintext symbols instead of equal partition. A modified algorithm with dynamically updated lookup table is presented in [20] and is illustrated in Fig. 2. Once a partition not matched with the target symbol is visited, all partitions mapped to the symbol of this partition are reassigned to a non-visited symbol. Therefore, the target symbol is eventually allocated with more partitions and fewer iterations are required to find it. This modification leads to a better compression performance, whereas the execution efficiency is comparable. A brief introduction of this algorithm is described as follows. For details, please refer to [20].

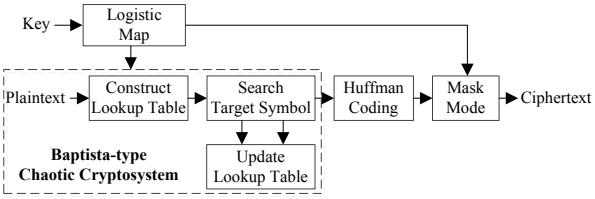

Fig. 2 Model of a modified chaos-based joint compression and encryption scheme [20].

Step 1: Construct the lookup table according to the probabilities of the occurrence of the source symbols.
Step 2: Sequentially encrypt each symbol in the plaintext sequence by searching it in the lookup table using a secret chaotic trajectory. If the target symbol is found, the number of iterations of the chaotic map is considered as the ciphertext. Otherwise, the partitions associated with the target symbol are reassigned to another symbol. The chaotic trajectory continues to search the target symbol in the updated lookup table until the target is found.
Step 3: When the current plaintext symbol has been encrypted, the lookup table is reinitialized. After that, the next symbol is encrypted using the same procedures in Step 2. These operations are repeated until all the symbols in the plaintext sequence have been processed.

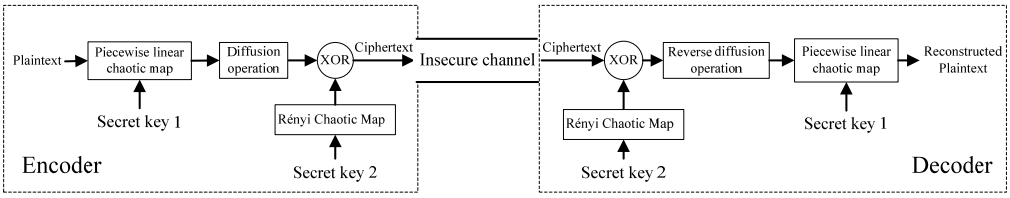

Fig. 3 Framework of the simultaneous source coding and encryption scheme using chaotic map [23].

Step 4: Compress the output of Step 3 using Huffman code. Small iterations are represented by a short code.
Step 5: Mask the output of Huffman code using random number bits generated by the secret chaotic trajectory.

2.3. Simultaneous Source Coding and Encryption Scheme Using Chaotic Map

In this scheme, a piecewise linear chaotic map is adopted in both compression and encryption, the resultant bits are further protected by a diffusion operation and an XOR operation controlled by another chaotic map [23]. Two secret keys are used, in which key 1 controls the secret models of piecewise linear chaotic map while key 2 is used as an initial value of the integer Rényi map for generating pseudo-random bits.

To introduce secrecy in the piecewise linear chaotic map, a secret key *KC* is used to cyclic-shift the position of the line segments whereas another secret key *KS* determines their directions. Both *KC* and *KS* are derived from secret key 1. The encoding procedures of this scheme can be briefly described as follows. The details can be found in [23].

Step 1: Read the input sequence and generate the corresponding statistical model. Determine the block size according to the entropy of source symbols in this block. Set the block counter i=1.
Step 2: Compress the source symbols in the i-th block with the variable-length AC approach using the secret chaotic map.
Step 3: The resultant bits obtained in Step 2 are further diffused by a global XOR operation and distorted by a left-shift operation.
Step 4: The diffused bits are then masked by the pseudo-random bits generated from an integer Rényi map.
Step 5: The secret keys *KS* and *KC* are periodically updated using the encrypted bits from the previous block.
Step 6: Set the block counter i=i+1. If all the source symbols have been encoded, output the entire encrypted bit stream. Otherwise, go to Step 2.

3. Experimental Results

In this section, three representative joint schemes [12, 20, 23] selected from different catalogues are studied experimentally. They are implemented in C language to compress 18 standard test files from the Calgary Corpus [24]. The parameter settings follow the suggestions in the published papers. For a fair comparison, all the algorithms adopt the static statistical model. All the simulations are performed on a personal computer with an E3 3.2GHz CPU and 2GB memory.

3.1. Compression Performance

The compression performance of the proposed schemes [12, 20, 23] are respectively shown in Table 1, where the compression ratio is obtained by dividing the ciphertext length by the plaintext length and then expressed in percentage. In particular, *BR* is the best compression ratio corresponding to the source entropy. In measuring the length of the ciphertext, the header information, such as the file length and the statistical data are included. Therefore, the compression ratios of all schemes are generally larger than *BR*. As shown in Table 1, the compression ratio is close to *BR* with an increase in file length, since the header information only occupies a small portion of ciphertext. Among the three schemes, the chaos-based adaptive AC scheme [12] achieves the best compression ratio that is very close to *BR* as it only distorts the mathematical model used in AC and does not affect the compression performance. Our scheme [23] ranks second as it needs to initialize the coding interval which induces compression loss when the number of bit shifts is not an integer. The Baptista-based joint scheme [20] gives the worst compression performance as the original plaintext redundancy is reduced by embedding the compression capability. In summary, the chaos-based

Table 1
Compression Ratio of Various Schemes

File	Size (byte)	*BR*	[12]	[20]	[23]
paper5	11,954	61.70%	64.05%	66.72%	65.19%
paper4	13,286	58.75%	60.61%	64.91%	61.70%
obj1	21,504	74.35%	77.99%	81.27%	79.28%
paper6	38,105	62.62%	63.39%	67.45%	64.50%
progc	39,611	64.99%	65.71%	70.10%	66.87%
paper3	46,526	58.31%	58.88%	64.37%	59.93%
progp	49,379	60.86%	61.43%	66.08%	62.52%
paper1	53,161	62.29%	62.85%	67.21%	63.97%
progl	71,646	59.63%	60.02%	64.61%	62.52%
paper2	82,199	57.52%	57.88%	63.31%	58.89%
trans	93,695	69.16%	69.50%	72.85%	70.72%
geo	102,400	70.58%	71.36%	78.35%	72.58%
bib	111,261	65.01%	65.24%	69.62%	66.39%
obj2	246,814	78.25%	78.61%	84.61%	79.93%
news	377,109	64.87%	64.96%	69.95%	66.11%
pic	513,216	15.13%	15.38%	27.46%	15.51%
book2	610,856	59.91%	59.97%	65.85%	61.03%
book1	768,771	56.59%	56.65%	62.15%	57.64%

adaptive AC scheme is better than ours by 0.13%–2.51% and the Baptista-based joint scheme by 2.67%–12.08%, respectively.

3.2. Security Analysis

The chaos-based adaptive AC scheme is shown to be vulnerable upon the chosen-plaintext attack [13] while our and the Baptista-based joint scheme are secure upon this attack as they possess two-level protections. The final encrypted bits from the chaotic cryptosystem are further masked by an XOR operation generated by another chaotic map, which can resist various attacks [2]. This XOR operation can also fix the security loophole of the chaos-based adaptive AC scheme.

4. Conclusion

In this paper, various chaos-based simultaneous compression and encryption schemes are reviewed. Three representative algorithms are picked for comparison. Their corresponding strengths in compression performance and security issues are reported with the support of simulation results.

References

[1] H. Kim, J. Wen, J. Villasenor, "Secure arithmetic coding," *IEEE Trans. Signal Process.*, vol. 55, pp. 2263-2272, 2007.

[2] J. Zhou, O.C. Au, P.H. Wong, "Adaptive chosen-ciphertext attack on secure arithmetic coding," *IEEE Trans. Signal Processing*, vol. 57, pp. 1825-1838, 2009.

[3] I. Witten, J. Cleary, "On the privacy afforded by adaptive text compression," *Comput. Secur.*, vol. 7, pp. 397-408, 1988.

[4] C. Wu, C. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Trans. Multimedia*, vol. 7, pp. 828-839, 2005.

[5] M. Grangetto, E. Magli, G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *IEEE Trans. Multimedia*, vol. 8, pp. 905-917, 2006.

[6] J. Wen, H. Kim, J. Villasenor, "Binary arithmetic coding with key-based interval splitting," *IEEE Signal Process. Lett.*, vol. 13, pp. 69-72, 2006.

[7] L.L. Duan, X.F. Liao, T. Xiang, **"**A secure arithmetic coding based on Markov model," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 16, pp. 2554-2562, 2011.

[8] J. Zhou, Z. Liang, Y. Chen, O.C. Au, "Security analysis of multimedia encryption schemes based on multiple Huffman table," *IEEE Signal Process. Lett.*, vol. 14, pp. 201-204, 2007.

[9] G. Jakimoski, K. Subbalakshmi, "Cryptanalysis of Some Multimedia Encryption Schemes," *IEEE Trans. Multimedia*, vol. 10, pp. 330-338, 2008.

[10] R.S. Katti, S.K. Srinivasan, A. Vosoughi, "On the security of randomized arithmetic codes against ciphertext-only attacks," *IEEE Trans. Inf. Forensic Secur.*, vol. 6, pp. 19-27, 2011.

[11] T. Xiang, C. Yu, J. Qu, X. Fu, "Cryptanalysis of secure arithmetic coding based on interval swapping," *Proceedings of 2011 CEWIT*, pp. 1-4, 2011.

[12] R. Bose, S. Pathak, "A novel compression and encryption scheme using variable model arithmetic coding and coupled chaotic system," *IEEE Trans. Circuits Syst I, Reg. Papers*, vol. 53, pp. 848-857, 2006.

[13] J. Zhou, O. C. Au, "Comments on 'A novel compression and encryption scheme using variable model arithmetic coding and coupled chaotic system'," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, pp. 3368-3369, 2008.

[14] B. Mi, X. Liao, Y. Chen, "A novel chaotic encryption scheme based on arithmetic coding," *Chaos Solitons & Fractals*, vol. 38, pp. 1523-1531, 2008.

[15] H. Li, J. Zhang, "A secure and efficient entropy coding based on arithmetic coding," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 14, pp. 4304-4318, 2009.

[16] H. Hermassia, R. Rhouma, S. Belghitha**,** "Joint compression and encryption using chaotically mutated Huffman trees," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 15, pp. 2987-2999, 2010.

[17] N. Nagaraj, P.G. Vaidya, K.G. Bhat, "Arithmetic coding as a non-linear dynamical system," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 14, pp. 1013-1020, 2009.

[18] K.W. Wong, C.H. Yuen, "Embedding compression in chaos-based cryptography," *IEEE Trans Circ & Sys II, Exp. Briefs,* vol. 55, pp. 1193-1197, 2008.

[19] H. Li, J. Zhang**,** "Embedding adaptive arithmetic coder in chaos-based cryptography," *Chin. Phys. B*, vol. 19, 050508, 2010.

[20] J.Y. Chen, J.W. Zhou, K.W. Wong, "A modified chaos-based joint compression and encryption scheme," *IEEE Trans Circ & Sys II, Exp. Briefs,* vol. 58, pp. 110-114, 2011.

[21] O.Y. Lui, K.W. Wong, J.Y. Chen, J.W. Zhou, "Chaos-based joint compression and encryption algorithm for generating variable-length ciphertext," *Appl. Soft. Comput.*, vol. 12, pp. 125-132, 2012.

[22] K.W. Wong, Q.Z. Lin, J.Y. Chen, "Simultaneous compression encryption using chaotic maps", *IEEE Trans Circ & Sys II, Exp. Briefs*, vol. 57, pp. 146-150, 2010.

[23] Q.Z. Lin, K.W. Wong, J.Y. Chen, "An enhanced variable-length arithmetic coding and encryption scheme using chaotic maps," *J. Syst. Softw.*, vol. 86, pp. 1384-1389, 2013.

[24] [Online]. Available: ftp://ftp.cpsc.ucalgary.ca/pub/projects/text.compression.corpus