A novel security arithmetic coding scheme based on nonlinear dynamic filter (NDF) with changeable coefficients is proposed in this paper. The NDF is employed to generate the pseudorandom number generator (NDF-PRNG) and its coefficients are derived from the plaintext for higher security. During the encryption process, the mapping interval in each iteration of arithmetic coding (AC) is decided by both the plaintext and the initial values of NDF, and the data compression is also achieved with entropy optimality simultaneously. And this modification of arithmetic coding methodology which also provides security is easy to be expanded into the most international image and video standards as the last entropy coding stage without changing the existing framework. Theoretic analysis and numerical simulations both on static and adaptive model show that the proposed encryption algorithm satisfies highly security without loss of compression efficiency respect to a standard AC or computation burden.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

The secrecy of transmitted or stored data is required in a variety of application, and multimedia data is not exception. The ease of processing, distributing and storing data on the Internet gave rise to many digital multimedia applications and services. However, the existing wired and wireless IP networks are open networks, and the data transmitted over these networks can be easily copied or modified. So, the emergence of digital rights management as an important filed for multimedia applications has attracted a lot of researchers.

Encryption is one of the major digital rights management enabling technology. Traditionally, to provide confidentiality, the data is encrypted using a stream cipher (RC4) or a block cipher (such as DES or AES) in some mode of operation for encryption. However, these schemes fail to take advantage of the additional design flexibility and potential computational simplifications. Some existing image encryption technologies have been developed in the spatial domain, such as [1–3], but these algorithms are harmful for correlation of two adjacent pixels in the spatial domain and this encryption scheme makes it difficult to compress and provide advanced functionalities such as conditional access, which are more easily implemented in a transformed domain. So, the issue of providing both compression and security simultaneously is growing more important given the increasing ubiquity of compressed media files and the common desire to provide security such as copyright protection, authentication and conditional access in association with these files.
The joint design of the compression and encryption parts potentially allows to provide a more flexible encryption of the data and the security of entropy coding has received attention in the literature. The entropy coder can improve the efficiency by doing both compression and encryption in a single step, without the need of an external encryption block. Among all the different entropy-coding methods, arithmetic coding stands out in terms of elegance, effectiveness and versatility, since it is able to work most efficiently in the largest number of circumstances and purposes. Traditional arithmetic coding provides essentially no security in the face of a chosen plaintext attack. These based approaches on AC, both adaptive and fixed-model, have been proven to be largely unsuitable for encryption by studies [4,5]. Adaptive modeling, in particular, which offers a huge model, can make decryption arbitrary difficult. But the scheme has shown to be vulnerable to organized cryptanalysis [6]. Moreover, these techniques result from reduced coding efficiency with respect to a standard AC. Nevertheless, based on the poor AC resynchronization, using arithmetic codes for encryption is motivated and has been studied in [7,8]. Ranjan Bose proposes a scheme which makes the structure of the model unpredictable and variable in nature using bitstream generated by the PRBG based on one coupled chaotic system [9]. Although compression is possible, this technology is not optimal for entropy coding. Multiple Huffman tables (MHT) is a scheme that performs both compression and encryption by using multiple statistical models (i.e., Huffman Coding Table) in the entropy coder [10]. To achieve both compression and confidentiality, the arithmetic coding with key-based interval splitting (KSAC) is proposed in [11], where the intervals in each iteration of an arithmetic code will be partitioned according to keys. While, in a traditional arithmetic coder, the intervals associated with each symbol are continuous. However, both MHT and KSAC are vulnerable to known plaintext attacks for low complexity [12].

More recently, a randomized arithmetic coding (RAC) system based on random swapping of the two intervals in a binary arithmetic coder is described by Grangetto et al., who utilizes this approach to encrypt JPEG2000 coded images [13]. In chaotic systems, the properties of initial-value sensitivity and parameter sensitivity make the initial-value and parameters suitable for encryption keys [14]. Based on the intrinsic nature of chaos, Bo. mi utilizes the logistic map as the pseudorandom bit generator, a novel chaotic encryption scheme based on RAC for image security in spatial domain is proposed in [15]. However, the RAC is not suitable for compressing the image in the spatial domain. It also should be noted that the trajectory of logistic map’s probability density function is not uniform distributed. And this property leads to small key space and its map parameter and initial value can be easily obtained in some conditions. So, this chaotic cryptosystems are vulnerable to cryptanalysis. Obviously, from the viewpoint of cryptography, the stream cipher which employs the logistic map is not sufficient secure [16]. Fortunately, the NDF chaotic map has been succeeded in applying chaotic secure communication [17] and the hash algorithm [18] due to its uniform distribution and large key space [19,20]. To enhance the security of cryptosystem, the plaintext is mapped into the chaos parameter space. The pseudorandom bitstreams depend on both initial value of NDF and the plaintext, which employs to flexible design the coefficients of NDF [19]. Theoretical analysis and computer simulations indicate that the proposed algorithm satisfies the security requirements without loss of compression performance.

The reminder of the paper is organized as follows. Section 2 briefly reviews the chaotic map, especially nonlinear dynamic filter. Section 3 details our proposed security arithmetic coding. In this section, its cryptanalysis is also discussed. While all of the methods described here can be applied for coding of source alphabets with any size, the authors address the case of binary systems here to simplify the discussion and illustrations. Section 4 provides experimental results on the performance of the proposed approach. The security issues are discussed in Section 5. Finally, in Section 6, we draw some conclusions and outline possible future developments.

2. $n$-Dimensional chaotic dynamic system

There are some problems that should be considered while designing good cipher based on chaotic systems, for example the large key space and the uniform distribution and ergodicity. But not all the chaotic map can satisfy these characters. Logistic map has no ideal uniform distribution and only one control parameter [16]. Although a piecewise linear chaotic map (PWLCM) is theoretically uniform distributed in the continuous phase space, the uniform distribution property of PWLCM will be collapsed in finite computing precision [21]. But, chaotic maps based on nonlinear dynamic filter can provide the large key space and the uniform distribution and will be given detailed description in the following.

Consider an $n$-dimensional ($n$-D) continue-value discrete-time nonlinear digital filter structure depicted in Fig. 1. The state space equation of the system is

\[\begin{align*}
\dot{x}(t) & = \sum_{i=1}^{n} a_i(t) x^{(i)}(t) + b_0(t) u(t) + b_1(t) v(t), \\
\dot{z}_i(t) & = \sum_{j=1}^{n} a_{ij}(t) x^{(j)}(t) + b_0(t) u(t) + b_1(t) v(t), \\
\end{align*}\]

Fig. 1. Block diagram of $n$-dimensional nonlinear dynamic filter.
\[
\begin{align*}
\begin{cases}
  z_1(t + 1) = h \circ \text{mod}(\sum_{i=1}^{n} c_i z_i + \phi), & \phi \in \Phi = R, \\
  z_k(t + 1) = z_{k-1}(t), & k = 2, 3, \ldots, n, \\
  y(t) = z_k(t + 1) \in I,
\end{cases}
\end{align*}
\]

where \( z = (z_1, z_2, \ldots, z_n)^T \in Z = \mathbb{F}^n \) denotes the vector of state variables. \( h(\cdot) \) is a piecewise linear map

\[
h : I \to I, \quad h(w) = m_k \cdot \omega + r_k, \quad \omega \in W_k \subseteq I, \quad k \in \{1, \ldots, M\}
\]

a modulo map

\[
\text{mod}(v) = v - 2 \left\lfloor \frac{v + 1}{2} \right\rfloor = v - 2 \cdot l,
\]

where \( v \) and \( l \) satisfy \( v \in [-1 + 2 \cdot I, -1 + 2 \cdot l], l \in G \).

Without loss of generality, \( I = [-1, 1] \). Because of the periodicity of modulo map, we restrict our consideration to the parameter interval \( \Phi = [-1, 1] \). The eigenvalues of Eq. (1) are denoted by \( \lambda_i, i = 1, \ldots, n \). Kelber [19, 20] has proved that Eq. (1) is an ergodic chaotic system and the state vector \( Z \) has the \( n \)-dimensional uniform probability density

\[
f(z, t) = f(z) = \begin{cases} 2^{-n}, & z \in \mathbb{F}^n, \\ 0, & z \notin \mathbb{F}^n, \end{cases} \quad t \in \mathbb{N}
\]

only if the following conditions are satisfied: (1) \( h(\cdot) \) is uniform distribution preserving; (2) the \( n \)-dimensional system cannot be decomposed into lower dimensional independent subsystems, that is \( |\lambda_i| \neq 1, i = 1, \ldots, n \); (3) the system parameters satisfy \( c_n \in Z \) and \( |c_i| > 1 \) The system parameters \( c_n \neq 0 \) can be chosen according to the synthesis method [19].

The \( n \)-th order nonlinear filter satisfying above conditions (Kelber conditions for short) is an \( n \)-D chaotic system with good cryptographic properties. Fig. 2(a) illustrated the trajectory of a second-order NDF map, under the conditions are that the system coefficients are \( c_1 = 5.7 \) and \( c_2 = 7 \), the initial values are \( z(0) = 0.6, z(0) = 0.2, z(2)(0) = 0.3 \) and the number of iteration is 10,000. From the trajectory of a nonlinear dynamic filter and its distribution function as illustrated in Fig. 2(b), we can see that the chaotic property of the NDF map is commendably preserved and it has the uniform probability density distribution.

3. Secure arithmetic coding by NDF with changeable parameters

Arithmetic decoding is very sensitive to errors in the compressed data, which tend to propagate throughout the decoded block; this otherwise undesirable property can be used to design a robust multimedia secure algorithm [13]. In fact, a single erroneous decoding step is able to cause an irreversible drift, thus making the data decoded further completely useless. One way of viewing the operation of arithmetic coding is that it generates output as a single real number which is uniformly distributed. The 0–1 balance property of stream cipher based on chaos provides generate maximum randomization in encryption process. These properties which provide good cryptography motivate us to implement the RAC based on NDF chaos. In this section, we will describe the detail of the security AC based on NDF, compression efficiency and cryptanalysis are also discussed.

3.1. Random binary sequences generated from NDF and plaintext

In cryptography, the independent and identically distributed binary random sequence is one of the most important components. Chaotic maps are usually employed to generate pseudorandom sequences for secure communication and cryptography. However, a majority of cryptosystems with keystreams independent of plaintexts are vulnerable under known plaintext attacks no matter how complicated the algorithms are [24]. To enhance the security of cryptosystem, the plaintexts should be considered when producing pseudorandom bitstreams. In [15], the plaintexts are mapped into the Logistic’s trajectory space (phase space), however, this technique does not improve the system security. It also should note that the trajectory of the logistic map is not uniform distribution, the parameter of the logistic map could be obtained. In this paper, the plaintexts are mapped into the NDF parameter space. The pseudorandom number generator is designed as follows.

\[
\xi_i \text{ is the system eigenvalue. It can be proven that Lyapunov exponents of system are } \lambda_i = \log |\xi_i|. \text{ For multidimensional systems, metric entropy is the sum of all positive Lyapunov exponents} [19]. \text{ Thus the metric entropy } H \text{ of NDF is given by}
\]

\[
H = \sum_{i>0} \lambda_i = \sum_{|\xi_i| > 1} \log |\xi_i| = \log \prod_{|\xi_i| > 1} |\xi_i|.
\]

If \( |\xi_i| > 1 \) for \( i = 1, 2, \ldots, n \); \( H \) will be simplified as

\[
H = \log \prod_{i=1}^{n} |\xi_i| = \log |c_n|.
\]
In a strict sense, the metric entropy is the information creation rate with respect to the generating partitions of phase space. For 2D NDF system, in order to get 2 bit information at one iteration, the metric entropy should be larger than 2. We translate the pending message to the corresponding ASCII numbers, by means of linear transform, and then get the eigenvalues in the following manners:

\[
\begin{align*}
\lambda_1 &= 3 + m_1/256, \\
\lambda_2 &= 3 + m_2/256,
\end{align*}
\]

where \( m_1 \) and \( m_2 \) is the ASCII value of original message. Then, the metric entropy is

\[
h = \sum_{i=1}^{2} \log |\lambda_i| > 1 + 1 = 2
\]

and so it satisfies the Kelber condition \cite{19}.

Fig. 2. The uniform property of a NDF. (a) Trajectory of a NDF; (b) its distribution function.
The NDF parameters (the filter coefficients) can be got by the synthesis method. In the 2D systems, it can be represented by
\[
\begin{align*}
    c_1 &= \xi_1 + \xi_2, \\
    c_2 &= -\xi_1 \xi_2.
\end{align*}
\]  

The pseudorandom number can be obtained by the output of NDF in the following manner:
\[
    r(i) = \begin{cases} 
    1, & y \geq 0.5, \\
    0 & \text{else}.
\end{cases}
\]

Although the coefficients of NDFs are substituted with time-variant ones, the output signal preserves the same properties (e.g., n-D uniform distribution) as the one with constant coefficients and is ergodic [17,18]. Furthermore, the security of this system is stronger than the one with fixed coefficients.

3.2. Secure arithmetic coding based on NDF with changeable parameters

Arithmetic coding is a nonlinear map [23], the entropy of output of arithmetic coding stream is optimized [22,26]. AC consists of two main parts, i.e., the statistical modeling and the AC compression engine. There are many advantages for separating the source modeling (probabilities estimator, accumulates statistics about the plaintext seen so far) and the encoding processing [22]. For example, it allows us to develop complex compression schemes without worrying about the details in the coding algorithm, and/or integrate them into different coding methods and implementations. Fig. 3 shows how the two processes can be separated in a complete system for arithmetic encoding and decoding.

During arithmetic coding process, statistical modeling plays an important role in data compression. The arithmetic coding leads to high performance compression only when model fits the statistical characteristics of data source. Clearly, when we interchange the ordering of symbols, it results in assuming a pseudostatistics of the incoming symbol stream. To retain the compression efficiency, correspondingly, we need to shuffle the probability of input symbols according to the order of the symbol, i.e., dynamically change the probabilistic model, prior to performing the arithmetic coding. We carry out the above two basic operations to make the arithmetic coding providing both compression and security simultaneously and the decoding completely key-dependent. Fig. 4 illustrated the procedure of the permutation. The concept of security arithmetic coding in the paper is depicted in Fig. 5. (This is a simple but efficient example.) The conventional arithmetic coding of the sequence “010010” is depicted in (a). The encoding of the same sequence using a randomized arithmetic coder is shown in (b). The ordering of intervals is determined according to a random and secret key stream (“010110”).

Finally, Fig. 6 gives a block diagram of the secure entropy encoding based on NDF. The system consists of NDF with changeable parameters to generate pseudorandom numbers which is employed in random arithmetic coding. The focus of the work presented in this paper has been incorporated the recent results of chaos theory, proven to be cryptographically secure, into arithmetic coding. The cryptography properties (0–1 balance and ideal correlation properties) of NDF as a PRBG are in favor of generating maximum randomization in the interval interchange process. In fact, a single erroneous decoding step is able to cause an irreversible drift, thus making the compressed data completely useless. The undesired property of AC in compression enhanced the security of the proposed scheme. It should be noted that we do not give the practical implementation aspects of AC for they have been well investigated and we only focus the difference between our proposed security AC and standard AC. Let \(p_0\) present the probability of the symbol “0” and \(p_1\) present the probability of the symbol “1”, the security entropy encoding based on NDF is operated as followings:

**Step1:** Initialize the parameters of 2D NDF and the initial values include input value, initial filter states. Here 2D NDF with changeable parameters is employed as the PRBG \(r_j\).

**Step2:** Implement the arithmetic encoding to encode the plaintext in the following way:

\[ \text{If } r_j = 1 \]
Exchange the value of $p_0$ and $p_1$;
Exchange the symbol "0" and the symbol "1";
else
Encode in a traditional way;
End

Step3: Update the NDFs parameters with the formula (8).
Step4: Go to step (2) until all inputs are encoded.

The above steps are mainly used in static model, for adaptive model, it only is needed to update the source modeling (probability) after Step2, before Step3. While the method described here can be applied for coding of source alphabets with
any size, we address the case of binary systems here to simplify the discussion and illustrations. The length of the pseudo-
random sequence which is utilized in encryption processing is equal with that of input binary symbol. Note that, if no ran-
domization is employed (i.e., is always equal to zero), RAC is identical to a standard arithmetic coding, so that RAC can also be
used to decode a nonencrypted data stream. Of course, one may decide to employ different amounts of randomization. In the
above, we maximize the randomization since the property of 0–1 balance of NDF with changeable parameters. The decryp-
tion algorithm is similar to the encryption algorithm, as illustrated in Fig. 7 which gives a block diagram of the secure en-
tropy decoding based on NDF. That is, for the encrypted sequence, firstly, decrypt the sequence using NDF system with the
same parameters and initial values as that used in encryption, and then exchange the symbol and its possibility bit by bit
separately if necessary, finally, we will get the original bit sequence.

3.3. Analysis of compression efficiency

Information theory [26] shows that the average number of bits needed to code each symbol from a statistical and
memoryless source cannot be smaller than its entropy. Let the probability of occurrence of the symbol “0” be \( p \) and that of
symbol “1” be \( 1 - p \). The Shannon’s entropy rate for this source is

\[
H = -p \log_2 p - (1 - p) \log_2 (1 - p).
\]  

Let \( S = \{s_1, s_2, \ldots, s_N\} \) be an arbitrary i.i.d binary sequence of \( N \) random symbols, assuming that the total overhead is a
positive number \( \sigma \) bits, the number of bits per symbol used for coding a sequence \( S \) should be bounded by

\[
B_S \leq \frac{\sigma - \log_2 (l)}{N},
\]  

where \( l \) represents the total interval at the end of the encoding process. It follows the iteration in arithmetic coding, the final
interval is

\[
l = \prod_{k=1}^{N} p(s_k).
\]  

And thus

\[
H = -p \log_2 p - (1 - p) \log_2 (1 - p).
\]  

Let \( S = \{s_1, s_2, \ldots, s_N\} \) be an arbitrary i.i.d binary sequence of \( N \) random symbols, assuming that the total overhead is a
positive number \( \sigma \) bits, the number of bits per symbol used for coding a sequence \( S \) should be bounded by

\[
B_S \leq \frac{\sigma - \log_2 (l)}{N},
\]  

where \( l \) represents the total interval at the end of the encoding process. It follows the iteration in arithmetic coding, the final
interval is

\[
l = \prod_{k=1}^{N} p(s_k).
\]  

And thus

\[
H = -p \log_2 p - (1 - p) \log_2 (1 - p).
\]  

Let \( S = \{s_1, s_2, \ldots, s_N\} \) be an arbitrary i.i.d binary sequence of \( N \) random symbols, assuming that the total overhead is a
positive number \( \sigma \) bits, the number of bits per symbol used for coding a sequence \( S \) should be bounded by

\[
B_S \leq \frac{\sigma - \log_2 (l)}{N},
\]  

where \( l \) represents the total interval at the end of the encoding process. It follows the iteration in arithmetic coding, the final
interval is

\[
l = \prod_{k=1}^{N} p(s_k).
\]  

And thus

\[
H = -p \log_2 p - (1 - p) \log_2 (1 - p).
\]  

Let \( S = \{s_1, s_2, \ldots, s_N\} \) be an arbitrary i.i.d binary sequence of \( N \) random symbols, assuming that the total overhead is a
positive number \( \sigma \) bits, the number of bits per symbol used for coding a sequence \( S \) should be bounded by

\[
B_S \leq \frac{\sigma - \log_2 (l)}{N},
\]  

where \( l \) represents the total interval at the end of the encoding process. It follows the iteration in arithmetic coding, the final
interval is

\[
l = \prod_{k=1}^{N} p(s_k).
\]  

And thus

\[
H = -p \log_2 p - (1 - p) \log_2 (1 - p).
\]  

Let \( S = \{s_1, s_2, \ldots, s_N\} \) be an arbitrary i.i.d binary sequence of \( N \) random symbols, assuming that the total overhead is a
positive number \( \sigma \) bits, the number of bits per symbol used for coding a sequence \( S \) should be bounded by

\[
B_S \leq \frac{\sigma - \log_2 (l)}{N},
\]  

where \( l \) represents the total interval at the end of the encoding process. It follows the iteration in arithmetic coding, the final
interval is

\[
l = \prod_{k=1}^{N} p(s_k).
\]  

And thus
\[ B_s \leq \frac{\sigma - \sum_{k=1}^{N} \log_2 p(S_k)}{N}. \]  

Assuming that the binary sequence consists of \( N_0 \) "0"s and \( N - N_0 \) "1"s. On average over an ensemble of such messages from the source, we have \( (N_0/N) \to p \), but individual messages may have different \( N_0 \). Also as \( N \to \infty \), for almost all individual messages we have \( (N_0/N) \to p \). Defining \( E(\cdot) \) as the expected value operator, the expected of bits per symbol is

\[ B = E(B_s) \leq \frac{\sigma - \sum_{k=1}^{N} E[\log_2 p(S_k)]}{N} = \frac{\sigma - (N_0 \log_2 p + (N - N_0) \log_2 (1-p))}{N} = \frac{\sigma}{N} - p \log_2 p - (1-p) \log_2 (1-p) = \frac{\sigma}{N} + H. \]

Since the average number of bits per symbol cannot be smaller than the entropy, we have

\[ H \leq B \leq H + \frac{\sigma}{N}. \]

And it follows that

\[ \lim_{N \to \infty} \{B\} = H. \]

Thus the proposed secure AC-coding achieves Shannon’s entropy rate of the source as the length of the message goes to infinity. The results mean that the proposed security arithmetic coding indeed achieves optimal compression performance.

3.4. Cryptanalysis of the proposed scheme

The resynchronization attack is not a known plaintext attack, toward which it is known [5,6] that arithmetic coding is not robust. For example, if the randomization is embedded in the statistical modeler, one could flood the encoder with zeros, and bring it into a known state. A similar attack could be carried (for example the all-zero sequence) the attacker could compare the AC-coded and RAC-coded sequences and attempt to infer the swapping decisions. But this case cannot happen in our proposed scheme. Even if the attacker get a keystream for all zero-sequence, but he does not obtain the initial value of the NDF chaotic system since a brute force attack would require the number of attempts equals to the key entropy, which is a problem of the number of precision bits allowed for the key. Furthermore, the keystream which is used in our scheme also depends on the plaintext and the model of the arithmetic coder depends on all text that has been coded since the initialization of the model, these make it immune to known plaintext attack and chosen plaintext attack.

It should be noticed that, since the proposed scheme encrypts a bit at one time, it can be seen as a stream cipher. But compared with the classical bitstream’s encryption, the nonlinear model is also introduced into our proposed security scheme during encoding processing. So, our scheme has strong strength in security. In a word, the scheme proposed in this paper has two advantages: (1) Both encryption and compression are achieved using a tool in our scheme, the joint design of the compression and encryption parts potentially allows to provide a more flexible encryption of the data, and hence more features. However, just encryption is obtained in the classical bitstream’s encryption. (2) For the encryption scheme via a stream cipher, it is not sufficient security for multimedia media in a coding and transmission environment since the initial portion of the data segment can be decrypted and decoded even if later portions have been lost (see, e.g., [25]). However, AC is well known to exhibits the fact that, in presence of errors, AC is well known to exhibit poor resynchronization capability [7]. And, only a synchronized decoder is able to interpret correctly the encoded sequence. The key of our proposed scheme is related with all the former plaintext and the arithmetic decoding processing also needs the former plaintext and the initial state. Our scheme provides a more robust data protection scheme than classical bitstream’s encryption scheme. Some researchers point out that the efficiency of the proposed scheme is expected to be worse than that of the standard stream cipher approach [12], but the security is enhanced, that is, the encryption strength is higher than the approach where one first compress the input data and then encrypts the results by a traditional stream cipher (such as Exclusive OR). In a particular case, if the probability of symbol "0" is equal with that of symbol "1", then the proposed scheme is equivalent to the classical bitstream’s encryption.

Let the probability of the symbol "0" be \( p_0 \) and that of the symbol "1" be \( p_1 \). If the binary source is i.i.d, in the static model, then the original binary and decrypted binary string with the a different key satisfied with the same probability distribution function, the bit error ratio (BER) is computed as following:

\[ p_e = 1 - p_0^2 - p_1^2. \]
In fact, BER is originally measured difference bit ratio between the two i.i.d. binary sequences with the same probability? However, in the adaptive model, during decryption (also decoding), resynchronization is very hard and it is impossible to capture the true probability of each symbol. Usually, since we cannot obtain the correct statistical model (probability estimator). The bit error ratio (BER) is computed as following:

\[
p_e = 1 - p_0 \times p_0' - p_1 \times p_1',
\]

where \( p_0' \) presents the probability of the symbol "0" in the decoded/decryption binary source and \( p_1' \) presents the probability of the symbol "1" in the decoded/decryption binary source. When \( p_0' = p_0, p_1' = p_1 \), the formulation (18) degenerates to formulation (17). So, we can treat formulation (17) as a special case of formulation (18).

4. Experiment results

The computer simulations are done to verify the performance of the proposed scheme. First, we choose a second-ordered nonlinear digital filter defined by the following:

![Figure 8](image-url)

(a) adaptive model.  
(b) static model

Fig. 8. The output (ciphertext) of encrypted binary source. The output of the proposed security AC shows that it has uniform distribution. During the AC coding process, the unit of output acts as byte, so the range of output is 0–255.
\[ y(n) = h \circ \text{mod} \left( c_1y(n-1) + c_2y(n-2) + \phi_0 \right), \]

where the piecewise linear map (PWL)

\[
h(w) =
\begin{cases} 
  \frac{w}{p}, & 0 \leq w < p, \\
  \frac{(w - p)/(0.5 - p)}{p - w} & p \leq w < 0.5, \\
  \frac{(1 - w - p)/(0.5 - p)}{0.5 \leq w < 1 - p}, \\
  \frac{(1 - w)/p}{1 - p \leq w < 1}, \\
  h(-w), & w < 0.
\end{cases}
\]

The initial values including input value \( \phi_0 = 0.6 \), initial filter states \( y(0) = 0.2 \), \( y(1) = 0.3 \). The initial parameters of NDF are \( c_1 = 5.7 \), \( c_2 = 7 \); the initial filter state value (the key) are \( (0.6, 0.2, 0.3) \).

The experiments for the proposed approach are conducted on a personal computer with an AMD Athlon(tm) 64 X2 Dual processor (2.61 GHz) and 1G RAM configured with Microsoft Windows XP and visual C++ 6.0. The size of binary source is 262,144 bits (32 KBytes) which is generated from the fastac toolbox it can be downloaded from the website (http://www.cipr.rpi.edu/~said/FastAC.html). We develop secure arithmetic coding based on the arithmetic coding toolbox for our experiments. The test binary data satisfies independent and identically distributed (i.i.d). Given a number of data symbols, it denotes several values of source entropy, and for each value it generates millions of pseudorandom source samples. It compares the decoded with the original data sample to make sure the code is correct or not. All arithmetic coding implementations have strict limits on the smallest probability they can support.

The output value of the proposed scheme has uniform distribution as illustrated in Fig. 8 for both on statistic model and adaptive model. In the proposed scheme, the encryption operation is combined with the compression process, and the decryption operation is combined with the decomposition process. As a good scheme, it should not affect compres-

---

**Table 1**

Compression efficiency of different information entropy.

<table>
<thead>
<tr>
<th>H</th>
<th>Entropy bound</th>
<th>Static model</th>
<th>Security AC</th>
<th>Adaptive model</th>
<th>Security AC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Traditional AC</td>
<td>Security AC</td>
<td>Traditional AC</td>
<td>Security AC</td>
</tr>
<tr>
<td>0.1</td>
<td>3277</td>
<td>3307</td>
<td>3307</td>
<td>3312</td>
<td>3312</td>
</tr>
<tr>
<td>0.2</td>
<td>6554</td>
<td>6599</td>
<td>6600</td>
<td>6603</td>
<td>6604</td>
</tr>
<tr>
<td>0.3</td>
<td>9831</td>
<td>9856</td>
<td>9856</td>
<td>9859</td>
<td>9860</td>
</tr>
<tr>
<td>0.4</td>
<td>13,108</td>
<td>13,100</td>
<td>13,100</td>
<td>13,104</td>
<td>13,104</td>
</tr>
<tr>
<td>0.5</td>
<td>16,384</td>
<td>16,417</td>
<td>16,417</td>
<td>16,420</td>
<td>16,420</td>
</tr>
<tr>
<td>0.6</td>
<td>19,661</td>
<td>19,740</td>
<td>19,740</td>
<td>19,742</td>
<td>19,743</td>
</tr>
<tr>
<td>0.7</td>
<td>22,938</td>
<td>23,009</td>
<td>23,009</td>
<td>23,011</td>
<td>23,012</td>
</tr>
<tr>
<td>0.8</td>
<td>26,215</td>
<td>26,251</td>
<td>26,251</td>
<td>26,254</td>
<td>26,254</td>
</tr>
<tr>
<td>0.9</td>
<td>29,492</td>
<td>29,530</td>
<td>29,530</td>
<td>29,532</td>
<td>29,533</td>
</tr>
</tbody>
</table>

---

**Fig. 9.** Comparison between the different AC versions and the ideal information entropy with different information entropy.
During the implementation of arithmetic coding based on finite precision, the length of interval during each iteration is not always the exactly same. It has a slight different as the interval position changes. Table 1 gives the compression results of different information entropy both on adaptive model and static model. In order to better visualize the results and their comparison to those in information entropy bound, Fig. 9 presents the difference of the compression results of the two proposed methods and the information entropy bound (which appears as a zero straight reference line). From Fig. 9, the most of encrypted and compression source is always the same with the traditional arithmetic coding both on adaptive model and static mode. From Table 1, the obtained bytes using adaptive model is a little higher than 2–5 bytes that of adaptive model with 32 KBytes binary source. The number of obtained bytes is not always the same with theoretical coded bytes based on data source entropy since probability of the test binary symbol is not exact for generating pseudorandom mechanism. From the above discussion, we may say that our proposed scheme unaffects on compression efficiency.

![Diagram](image-url)

**Fig. 10.** The encryption key sensitivity (initial values of NDF) of i.i.d. binary source both on static and adaptive model. The key difference scales down by ratio of $10^{-1}$.
5. Security analysis

A good encryption scheme should resist all kinds of known attacks, it should be sensitive to the secret keys during implement of both encryption and decryption, and the key space should be large enough to make brute-force attacks infeasible. In the proposed chaotic encryption scheme, the pseudorandom sequences are generated from the NDF system and used to encrypt plaintext in the stream manner. The security depends on the randomness of the pseudorandom sequences and the key space. They are analyzed and tested, respectively as follows.

5.1. Key sensitivity

The change of a single bit in the secret key should produce a completely different encrypted ciphertext. The entropy of the test data source is \( H = 0.85 \). Fig. 10 gives the encryption key sensitivity (initial values of NDF) of i.i.d. binary source both on static and adaptive model. From Fig. 10, The encryption with a slightly different parameter or initial values will generate big different cipher text, about 50%; hence the proposed entropy encryption scheme is highly key sensitive.

The decryption key sensitivity of i.i.d. binary source is also tested in our experiments, as described in Tables 2 and 3. Table 2 presents the decryption key sensitivity of i.i.d. binary source with \( H = 0.85 \) both on static and adaptive model. Table 3 shows that the BER of the different Shannon information entropy of the test data source from 0.1 to 0.9. In our experiments, the data source entropy is \( H = 0.85 \), the probability of symbol ‘0’ is \( p_0 = 0.7228, p_1 = 0.2772 \) the bit error ratio can be obtained by \( p_e = \frac{1}{C_0} \left( p_0^2 - p_0^2 \right) \) from the figures in Tables 2 and 3, we can see that the \( H \) is not fixed and varies in a large scope. The BER in our experiments are very close to the theoretical results.

### Table 2
The decryption key sensitivity of i.i.d. binary source with \( H = 0.85 \).

<table>
<thead>
<tr>
<th>The difference key</th>
<th>Static model</th>
<th>Adaptive model</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p_0 )</td>
<td>0.3965</td>
<td>0.3748</td>
</tr>
<tr>
<td>( H )</td>
<td>0.7810</td>
<td>0.7538</td>
</tr>
<tr>
<td>p_0</td>
<td>0.3760</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>0.9972</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3
The decryption key sensitivity of different i.i.d. binary source entropy.

<table>
<thead>
<tr>
<th>H</th>
<th>Static model</th>
<th>Adaptive model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>p_0</td>
<td>Theoretical BER</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9870</td>
<td>0.0257</td>
</tr>
<tr>
<td>0.2</td>
<td>0.9696</td>
<td>0.0601</td>
</tr>
<tr>
<td>0.3</td>
<td>0.9476</td>
<td>0.1118</td>
</tr>
<tr>
<td>0.4</td>
<td>0.9211</td>
<td>0.1463</td>
</tr>
<tr>
<td>0.5</td>
<td>0.8900</td>
<td>0.1970</td>
</tr>
<tr>
<td>0.6</td>
<td>0.8533</td>
<td>0.2473</td>
</tr>
<tr>
<td>0.7</td>
<td>0.8092</td>
<td>0.3071</td>
</tr>
<tr>
<td>0.8</td>
<td>0.7565</td>
<td>0.3621</td>
</tr>
<tr>
<td>0.9</td>
<td>0.6848</td>
<td>0.4333</td>
</tr>
</tbody>
</table>

5. Security analysis

A good encryption scheme should resist all kinds of known attacks, it should be sensitive to the secret keys during implement of both encryption and decryption, and the key space should be large enough to make brute-force attacks infeasible. In the proposed chaotic encryption scheme, the pseudorandom sequences are generated from the NDF system and used to encrypt plaintext in the stream manner. The security depends on the randomness of the pseudorandom sequences and the key space. They are analyzed and tested, respectively as follows.

5.1. Key sensitivity

The change of a single bit in the secret key should produce a completely different encrypted ciphertext. The entropy of the test data source is \( H = 0.85 \). Fig. 10 gives the encryption key sensitivity (initial values of NDF) of i.i.d. binary source both on static and adaptive model. From Fig. 10, The encryption with a slightly different parameter or initial values will generate big different cipher text, about 50%; hence the proposed entropy encryption scheme is highly key sensitive.

The decryption key sensitivity of i.i.d. binary source is also tested in our experiments, as described in Tables 2 and 3. Table 2 presents the decryption key sensitivity of i.i.d. binary source with the information entropy to \( H = 0.85 \), for both on static and adaptive model. Table 3 shows that the BER of the different Shannon information entropy of the test data source from 0.1 to 0.9. In our experiments, the data source entropy is \( H = 0.85 \), the probability of symbol “0” is \( p_0 = 0.7228, p_1 = 0.2772 \) the bit error ratio can be obtained by \( p_e = \frac{1}{C_0} \left( p_0^2 - p_0^2 \right) \) from the figures in Tables 2 and 3, we can see that the \( H \) is not fixed and varies in a large scope. The BER in our experiments are very close to the theoretical results.
which are computed from formula (17) (static model) and formula (18) (adaptive model). This means that the decrypted bit-string with the wrong key has good statistical property and much information. The proposed entropy encryption scheme is highly key sensitive and has achieved good security performance.

5.2. Key space

Key space size is the total number of different keys that can be used in the encryption. A good encryption scheme should be sensitive to the key space and should be large enough to make brute-force attacks infeasible. In our scheme, the secret key SK which includes initial values of NDF and the control parameter $p$ in the piecewise linear map is given with $SK = \{\phi_0, y(0), y(1), p\} = \{0.6, 0.2, 0.3, 0.35\}$. Each component of the key scales down from initial values by a ratio of $10^{-1}$ individually, and the corresponding ciphertext various ratio is shown in Fig. 10. The abscissa is scaled by negative logarithm. The

---

**Fig. 11.** Autocorrelation test of the ciphertext.
5.3. Analysis of ciphertext

The output value of the proposed scheme has uniform distribution as illustrated in Fig. 8 for both on statistic model and adaptive model. Therefore, attacks by analyzing the static property of ciphertext are prevented in our scheme. The output value of the proposed scheme has ideal auto-correlation property, as illustrated in Fig. 11 for both on statistic model and adaptive model. If the output value is represented in binary format, then it has good 0–1 balance both on statistic model and adaptive model. For adaptive model, the ratio of symbol “0” and “1” is 0.9931 and for static model the ratio of symbol “0” and “1” is 1.0032. The balance of 0–1 also shows that the proposed is entropy optimal. The proposed cryptosystem possesses the property of a ciphertext without any pattern of the plaintext. These correlation analysis demonstrate that the proposed chaotic encryption algorithm satisfy high-level security.

6. Conclusion

A novel encryption paradigm for the security of transmitted or stored data based on NDF chaos map is presented in this paper. By exploiting the structure of entropy coder, the proposed scheme demands very low complexity and very high flexibility, as it can be easily adapted to any block length. We also demonstrate from information theory's point of view that our scheme does not weaken the statistical randomness of compressed bit stream. Being efficient and secure, our proposed scheme is suitable to encrypt and decrypt multimedia data in highly demanding applications where large amount of data needs to be proposed. It can be integrated into most of image/video compression standard at negligible cost without changing the existing framework or adding external encryption block.

Acknowledgement

The authors would like thank the anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China under Grant No. 60572027, by the Outstanding Young Researchers Foundation of Sichuan Province Grant No. 03ZQ026-033, by the Program for New Century Excellent Talents in University of China under Grant No. NCET-05-0794.

References