Random Chaotic Arithmetic Coding for Video Compression and Encryption

K. John Singh, Ashutosh Garg, Suruchi Sharma and Tanmay Dubey

School of Information Technology and Engineering VIT University, Vellore, Tamil Nadu, India

Abstract: This paper studies the joint video compression and encryption, which gains good attention in past couple of years due to its reduced computational complexity. In this paper, we present a multimedia data encryption scheme based on arithmetic coding, which we refer to as Random Chaotic Arithmetic Coding (RCAC). In RCAC, we used eight different chaotic maps to perform encoding. Thus achieving equivalent compression performance as in arithmetic coding. The choice of map is depend on the value of key generated randomly. Different map used in RCAC only scramble the intervals in which the encoded word must lie, without making changes to width of that interval, thus allowing encryption without sacrificing any coding efficiency. We have also presented security enhancement of above method with little modification. A key sensitivity analysis is presented along with security analysis which proves the relevancy of scheme.

Key words: Arithmetic Coding - Chaotic Arithmetic Coding - Compression - Encryption - Joint Video Compression and Encryption

INTRODUCTION

In recent years, due to development of network technology and multimedia technology, multimedia data, especially video data are used all over the globe widely, thus it becomes a necessity to provide security to multimedia contents. Hence to protect video data, Encryption appears to be an efficient way, also videos are usually large in size and bulky, thus requires Compression to speed up the transmission time over the internet applications. We require both compression and encryption of video data before transmitted over internet. In this paper, unlike traditional approach of Encryption after compression, we used the Joint Video compression and encryption (JVCE) model which got good attention in last couple of years, since it helped in reducing the overall computational complexity of video processing for internet applications and services. A JVCE based on Arithmetic Coding and Chaotic map is proposed and also some security enhancements features are presented to make this scheme more secure against cryptanalysis attacks.

Arithmetic Coding (AC) was introduced in 1979, by J.J. Langdon [1] it is entropy coding, which involves associating a sequence of symbols with a position in the range [0,1) and is usually implemented recursively. Later, Grangetto et al [2] introduced an efficient encryption technique for arithmetic coding by altering the sequence in random manner known as the Randomized Arithmetic Coding (RAC) technique. RAC needs a key of length 1-bit per encoded symbol. Wen and Kim et al [3] presented a generalization of this procedure, called as key-Splitting Arithmetic Coding (KSAC) where a key is used to split the intervals of an arithmetic coder.

Recently many efforts have been made towards representing joint architecture of encryption and compression [4] modules (basically with entropy coding techniques such as arithmetic coding) in order to provide more flexible encryption of data. These techniques allow encryption at little or no computational complexity and also preserve the format compatibility of compressed bit stream.

Alvarez, F. Montoya, M. Romera [5] introduced chaotic cryptosystem which later improved by Shujun Li, Xuanqin Mou, Yuanlong Cai [6]. Also Mohamed I. Sobhy and Alaeldin R. Shehata in [7] showed that many techniques have been applied to all the published chaotic encryption systems and they observed that all those systems are very easy to broken. In this paper we...
introduce noble encryption scheme which uses the randomly generated secret key to encrypt the data and can be implemented easily.

**Arithmetic Coding as a Chaotic System:** The paper presented by Nagraj et al [8] derives the new method of arithmetic coding and chaotic linear maps (which was named as GLS coding). They developed a new method for deriving skewed maps, which is skewed with a non-linear parameter for encryption and compression of data. The above approach has some disadvantages, which makes the scheme prone to attack by attackers, they are:

- If we put wrong value of skew parameter then it lead to imperfect reconstruction and also have chance of getting completely random output. If Value of á is closely related to skew parameter then we may guess some initial part of binary string correctly.
- There is also possibility of iteratively guessing the value of á by studying known plaintext attack. If they are: The decoding part is also simple to achieve. Let us take encoded word to be 0.625. Iteration on the chaotic map mentioned in Equation (1) gives the values which makes the scheme prone to attack by attackers, arithmetic coding.

The equivalent method of chaotic maps to that of arithmetic coding is explained briefly with an example:

Suppose we are encrypting the binary string ‘100’ which is of length 3 bits. For Arithmetic Coding, let p (probability for occurrence of symbol ’0’) is initially selected to 0.6. For arithmetic coding, we will take the interval [0.6, 1) in first iteration to encode ‘1’, interval [0.6, 0.84) to encode the ‘10’ and interval [0.6, 0.744) to encode the ‘100’ thus a number lying in the interval [0.6, 0.744) represents the Arithmetic Code for binary string ‘100’.

Now, Skewed binary map can be defined with the following equations:

\[
y = \begin{cases} 
  x/p & \text{when } x \leq p \\
  (x-p)/(1-p) & \text{when } x > p
\end{cases}
\]

The decoding of symbol can be defined as follows:

\[
\text{Decode} \begin{cases} 
  '0' & \text{when } x \leq p \\
  '1' & \text{when } x > p
\end{cases}
\]

Forward iteration of above chaotic map is equivalent to decoding the arithmetic code whereas Back iteration of this map is equivalent to the arithmetic encoding, which can be defined by the following equation:

\[
x = \begin{cases} 
  py & \text{when } '0' \\
  (1-y)y + p & \text{when } '1'
\end{cases}
\]

To encode binary string ‘100’ with the P value preselected to 0.6. We substitute value of P to get the chaotic map. In this system, Encoding of data is done in reverse order, from end to the beginning thus we first encode ‘0’ and get the interval [0, 0.6) then next we encode ‘00’ to get the interval [0, 0.36) and finally we encode ‘1’ to get the interval which is corresponding to ’100’ as [0.6, 0.744). So we get the same interval as the arithmetic code.

The decoding part is also simple to achieve. Let us take encoded word to be 0.625. Iteration on the chaotic map mentioned in Equation (1) gives the values [0.625, 0.0625, 0.1041] in two iterations. Decoding it using the rule given in Equation (2) gives us the original sequence [100] or binary string ‘100’.

**Chaotic Arithmetic Coder:** In the previous section we explained how arithmetic coding can be viewed as re-iteration on skewed binary map. Thus we can use eight equivalent modes of skewed binary maps which can be used for iteration. They are shown in Fig. 1. These modes can be differentiated on the basis of input which is mapped into the chaotic function. The maps only differ in the interval of symbol ’0’ or ’1’ in which the arithmetic code must lie. All the modes are equivalent on the basis compression efficiency because the width of interval remains the same in all the modes.

Here we have defined the generalized equation for skewed binary map as:

\[
y = \begin{cases} 
  a_1x + b_1 & \text{when } x \leq k \\
  a_2x + b_2 & \text{when } x > k
\end{cases}
\]

\[
\text{Decode} \begin{cases} 
  '0' & \text{when } x \in [i_1,i_2] \\
  '1' & \text{when } x \in [i_3,i_4]
\end{cases}
\]

\[
x = \begin{cases} 
  c_1y + d_1 & \text{when } '0' \\
  c_2y + d_2 & \text{when } '1'
\end{cases}
\]

Where \( a, b, c \) and \( d \) values is used to compute the slope of chaotic map. These values can be pre-evaluated to a tabular form for direct lookup at time of implementation as shown in the Table 1. \( a_1 = A1(i), a_2 = A2(i), b_1 = B1(i), b_2 = B2(i), c_1 = C1(i), c_2 = C2(i) \).
Fig. 1: Represents the eight different modes of the chaotic binary map with p=0.6

Table 1: Parameter list for the eight possible choices of chaotic encoder

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
<th>(f)</th>
<th>(g)</th>
<th>(h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>1/p</td>
<td>1/p</td>
<td>-1/p</td>
<td>-1/p</td>
<td>1/(1-p)</td>
<td>1/(1-p)</td>
<td>-1/(1-p)</td>
<td>-1/(1-p)</td>
</tr>
<tr>
<td>B1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A2</td>
<td>1/(1-p)</td>
<td>-1/(1-p)</td>
<td>-1/(1-p)</td>
<td>1/(1-p)</td>
<td>1/p</td>
<td>-1/p</td>
<td>-1/p</td>
<td>1/p</td>
</tr>
<tr>
<td>B2</td>
<td>-p(1-p)</td>
<td>1/(1-p)</td>
<td>1/(1-p)</td>
<td>-p(1-p)</td>
<td>(p-1)/p</td>
<td>1/p</td>
<td>1/p</td>
<td>(p-1)/p</td>
</tr>
<tr>
<td>C1</td>
<td>p</td>
<td>p</td>
<td>-p</td>
<td>-p</td>
<td>p</td>
<td>-p</td>
<td>-p</td>
<td>p</td>
</tr>
<tr>
<td>D1</td>
<td>0</td>
<td>0</td>
<td>p</td>
<td>p</td>
<td>1-p</td>
<td>1</td>
<td>1</td>
<td>1-p</td>
</tr>
<tr>
<td>C2</td>
<td>1-p</td>
<td>p-1</td>
<td>p-1</td>
<td>p-1</td>
<td>1-p</td>
<td>1-p</td>
<td>1-p</td>
<td>p-1</td>
</tr>
<tr>
<td>D2</td>
<td>p</td>
<td>1</td>
<td>p</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1-p</td>
<td>1-p</td>
</tr>
<tr>
<td>I1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(1-p)</td>
<td>(1-p)</td>
<td>(1-p)</td>
<td>(1-p)</td>
</tr>
<tr>
<td>I2</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>I3</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1-p</td>
<td>1-p</td>
<td>1-p</td>
<td>1-p</td>
</tr>
<tr>
<td>K</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>1-p</td>
<td>1-p</td>
<td>1-p</td>
<td>1-p</td>
</tr>
</tbody>
</table>

\[ d_i = D1(i), d_2 = D2(i), i_1 = I1(i), i_2 = I2(i), i_3 = I3(i) \] and \( i_4 = I4(i) \) and \( i \) can be any of between 1 to 8 depending on chaotic map chosen. Table 1 provides the value for these parameters for all eight chaotic maps designate as (a)-(h).

**Implementation Efficiency:** In normal binary arithmetic coder, at each iteration, we updated the starting interval \([I_-, I_+]\) at one end. When encoding a ‘0’ the final interval becomes \([I_+ + P (I_+ - I_-), I_+]\) while encoding a ‘1’ the final interval becomes \([I_+, I_+ + P (I_+ - I_-)]\). Thus at each iteration it perform one multiplication and two addition operations. Also, While decoding of binary arithmetic encoded message involves updating the interval \([I_-, I_+]\) at one end depending on the value of last decoded symbol (either ‘0’ or a ‘1’). Thus at each iteration, it again performs one multiplication and two addition operations.

For Chaotic Arithmetic Encoder, at each iteration it require updating of both the ends of interval using a linear transformation \(x = cy + d\). Thus for encoding, at each iteration it perform two multiplications and two additions operations. Whereas achieving decoding is simple comparatively, which involves iteration through the chaotic map using the linear transformation \(y = ax + b\). It involves a multiplication and an addition operation. There are also some additional table lookups involved in chaotic coding according to choose chaotic map at every iteration which can be easily implemented with no cost overhead.

Thus, by above discussion we have derived that CAC perform more computations compared to BAC encode but CAC decode perform less computations compared to BAC decode. Also all eight maps provide the same final width of interval for the encoded word to lie in, but the interval differs for each maps.
Applications: The CAC can be used as a joint compression-cum-encryption technique for encrypting the Data. It is preferably used for encryption and compression of multimedia data such as digital video. CAC can also be used as either full or selective encryption of multimedia data. For performing full encryption, the entire multimedia data have to pass through CAC encoder whereas in selective encryption only the sensitive or selected parts of data have to pass through CAC encoder and rest data simply surpassed.

Random Chaotic Arithmetic Coding: RCAC is modified version of CAC, which perform joint compression and encryption of video data. Encoder use the key generated by pseudo-random number generator (PRNG) which takes seed as input to generate random key, the modulo operation is performed on key with 8 to get value between 0 to 7, which represent 8 different chaotic map.

The Component shows in Fig. 2, comprises of two modules: 1) RCAC encoder/decoder, 2) Key Generator. The input to the component is Plaintext which goes to the RCAC encoder as binary bit stream, which is encoded using chaotic map choose based on the key stream generated by key generator. The output of RCAC encoder is Cipher-text.

Key Generator included in the above component uses PRNG (Pseudo Random Number Generator) to generate random key and thus provides design flexibility. The input to the key generator is Seed which also serve as public key, known by the receiver in order to deduce the encrypted data. Thus, Seed must be unique and act as only private information in enhanced system.

The Working of Proposed Scheme Is Explained by the Following Algorithm:

Step 1: Input the Plaintext PT to the component which input as a binary stream to the RCAC Encoder.

Step 2: Seed which act as a Public Encryption Key is provided to Key Generator, which generate a random Keys K.

Step 3: Here K represents key -bit stream for RCAC encoder, Modulo operation is performed on key with 8 to get value between 0 to 7, which is used to choose a map among 8 different chaotic map.

Step 4: Data is encoded based on chosen map by computing equation and table lookup.

Step 5: The finally generated bit stream obtained by component is Cipher text CT.

Most of the Encryption algorithm makes use of the secret key. But the ideal encryption algorithm should make use of secure key exchange protocol which increases the security factor of scheme.

Enhancing Security: To enhance the security of RCAC technique. We presented two simple modification of above mentioned method using XOR functionality.

First, using XOR we alter the sequence of symbol in Plaintext by perform XOR operation between the plaintext message and key and provide resultant data to RCAC encoder to generate the final encrypt text. Also for Decoding first encrypted data is provided to RCAD and then XORed with key to reconstruct the encoded data.

Second modification, in which the Plaintext message is first provided to RCAC encoder then XOR operation, is performed between resultant data and key to provide final Encrypted text. Also, for decoding encrypted data is first XORed with key and resultant data is provided to RCAD to reconstruct the encoded data. These modifications shown in Fig. 3(a) and Fig. 3(b) greatly enhances the security performance without compromising the coding efficiency with slightly increase in computation.
Fig. 3(a-b): RCAC security enhancements

Also we can apply both mentioned modification together by including two XOR operation, before and after of RCAC/RCAD to encode and decode the data which also increased the computation on cost of security.

RESULTS

Security Analysis: Cipher-text only attack: Cipher text only is a type of attack in which it is assumed that attacker has access to a set of limited cipher texts. During this attack, the attacker achieves success when either he assumes the key or recognizes the pattern of encrypted data. In RCAC, we are encrypting the data by one of eight different Chaotic Maps, which can be determined by randomly generated key and hence the output pattern becomes unrecognized by the attacker.

Chosen Plain Text Attack: Chosen plain text attack is a type of attack in which the attacker has the potential to get the random plaintexts and with the help of which they deducing the corresponding cipher texts. By gaining more information, key can be revealed from chosen plain text.

In RCAC, key is generated randomly by providing seed, which is known by the sender and the receiver only. Thus, the attacker is unable to presume the key, since key is separated from the cipher text.

Known Plain Text Attack: Known Plain text attack is a type of attack in which several plaintexts / cipher texts pair is available to the attacker. By keenly studying the pair, attacker is able to recognize the pattern of encrypted data. In RCAC, the output data is both compressed and encrypted using Chaotic Function which scrambles the sequence. Thus, the probability of determining the pattern by the attacker is very negligible.

CONCLUSION

This paper presents a Joint Video Compression and Encryption scheme which reduces the computational complexity involved with traditional approach of encryption after compression. The proposed RCAC scheme uses an interpretation of Arithmetic Coding using chaotic maps. The proposed security enhancements allow us to build a video encryption scheme resistant to known attacks. The compression performance of RCAC scheme and the Binary Arithmetic Coder is same. RCAC scheme has higher computational costs on encoder side but reduced computational requirements on decoder (than BAC) with added security feature, which makes it favourable for server-client video distribution model which require easy decoding on client side and extended computations on server. The secure multicast idea presented in this paper can be further developed to be deployed in modern systems. The approach presented in this paper provides low-cost, low-overhead, property preserving encryption of videos.

REFERENCES


