Research Journal of Applied Sciences 8 (10-12): 494-500, 2013
ISSN: 1815-932X

Enhanced Randomized Arithmetic Coding Technique for Joint Compression and Encryption of Video Data

K. John Singh and R. Manimegalai
[1]School of Information Technology and Engineering, VIT University,
Vellore, Tamil Nadu, India
[2]Department of Computer Science and Engineering,
Park College of Engineering and Technology, Coimbatore, Tamil Nadu, India

Abstract: Joint video compression and encryption techniques gain good attention in recent years due to its reduced computational complexity, memory utilization and execution time. In this study, a modified encoding algorithm called Enhanced Randomized Arithmetic Coding (ERAC) is proposed for joint compression and encryption. The Randomized Arithmetic Coding algorithm (RAC) is modified to integrate encryption with compression. In ERAC, the video data is encrypted using a secret key during compression. It is simple to implement and has less computational overhead. The proposed ERAC technique has two modules, namely, Randomized Arithmetic Coding (RAC) module and the XOR module. The experimental results achieved and analysis of the proposed solution shows that the enhancement done over RAC provides high security without compromising the compression ratio and speed.

Key words: Arithmetic Coding (AC), Randomized Arithmetic Coding (RAC), Enhanced Randomized Arithmetic Coding (ERAC), video, data

INTRODUCTION

With the increasing usage of multimedia data and technology over the globe, it becomes necessary to provide security to video contents. Encryption techniques are used to protect video data from security violations. In general, videos are larger in size and hence require compression in order to reduce the size. The reduced size in turn, speeds up transmission over the Internet. There are various efficient encoding techniques available in the literature for video compression (Huffman, 1952; Moffat *et al*., 1995; Kim *et al*., 2007). Basically, the compression process packs video data into smaller space. The video data consumes large amount of memory without compression and must be compressed before it is encrypted, transmitted, stored or put up on the web. There are two types of compression, namely, lossy compression and lossless compression.

As the name indicates, the compressed file loses some amount of information when lossy compression is employed. As some amount of information is lost, lossy compression leads to quality degradation when compared to lossless compression. However, the biggest advantage of lossy-compression is reduced file size after compression. Lossy-compression is employed in situations where one can afford to lose relatively large amount of data with small compromise in quality such as streaming media and internet telephony. In lossless-compression, no information is lost but there is not much difference in file sizes before and after compression. This seems to be contradictory to the goal of compression. However, lossless compression techniques are used in applications where the file size and memory consumption are low priority.

It is mainly used for text and data files where high accuracy is required such as bank records, employee details and text articles. Entropy coding is one of the lossless compression schemes which compress the data by replacing each fixed length input symbol by the corresponding variable length prefix-free output codeword. There are two types of entropy coding techniques used for data compression, namely, Huffman Coding (HC) and Arithmetic Coding (AC).

Huffman coding is one of the basic compression technique proposed by Huffman (1952). Arithmetic Coding (AC) was proposed by Moffat *et al*. (1995). When compared to Huffman coding, arithmetic coding provides high coding efficiency and is preferred for compression of video data (Witten *et al*., 1987; Wen *et al*., 2006).

Later, Grangetto *et al*. (2004) modified the Arithmetic Coding and introduced Randomized Arithmetic Coding

Corresponding Author: K. John Singh, School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu, India

(RAC) technique. In this study, an enhanced version of Randomized Arithmetic Coding (RAC) is proposed which provides improved security.

Literature review: Huffman (1952) proposed an entropy encoding algorithm called Huffman coding for lossless data compression. It uses a variable length code table for encoding symbols or characters in a file. The codes are derived based on the estimated probability of occurrence for each possible value of the source symbol. It is also called as prefix-free code because any of the bit-string representing one particular symbol will not be the prefix of another bit-string representing some other symbol. Huffman coding is employed in static applications such as FAX, electronic books, ZIP files, JPEG, MPEG. Arithmetic coding is a technique which converts a given probability distribution into an optimal code and is commonly used in compression schemes. Main advantages of arithmetic coding are its optimality and its inherent separation of coding and modeling. Pure arithmetic coding is optimal for a stochastic data source whose probabilities are exactly known but it relies on relatively slow arithmetic operations like multiplication and division. Howard and Vitter (1994) proposed an efficient implementation of Arithmetic Coding which provides an effective mechanism for removing redundancy in encoding of data. The proposed solution by Howard and Vitter (1994) uses a table lookup procedure as a fast alternative to the slow arithmetic operations. Probability models are used to design the arithmetic coder and the implementation is speed up by parallel processing.

Though arithmetic coding is better than Huffman codes, it is prone to various attacks such as known and chosen plaintext attack. In (John, 1999) a preliminary analysis on the use of arithmetic coding as encryption scheme is done by Cleary *et al.* (1999). In arithmetic coding, an entire source sequence is mapped to a single code stream. Therefore, a single error in an arithmetic code stream often causes a lot of trouble at the decoder. Moo and Wu (1999) have proposed a solution for decoding an arithmetic code stream when an initial segment of that code stream is unknown. Decoding under these conditions is called as desynchronizing an arithmetic code (Moo and Wu, 1999). There are two variants of arithmetic coding, namely, Randomized Arithmetic Coding (RAC) and Key-based Secured Arithmetic Coding (KSAC). The RAC algorithm is proposed by Grangetto *et al.* (2004) in which randomization is combined with traditional arithmetic coding by randomly swapping the intervals of least and most probable symbol. The selective encryption technique proposed by Grangetto *et al.* (2006) has less computational complexity when compared to RAC.

In (Grangetto *et al.*, 2006) only selected important portions of the data are encrypted. The KSAC is proposed by Wen *et al.* (2006) in which a key is used to "split" the interval before encoding new symbol thus allowing compression and encryption simultaneously. Later in the same year, a new encryption algorithm was proposed by Socek *et al.* (2007) in which encryption process preserves the spatial correlation. In 2007, research was conducted by Xie and Kuo (2007) on encrypting multimedia data using joint randomized entropy coding and rotation in partition bit stream which incurs very less computational and implementation cost.

A thorough study of Secure Arithmetic Coding (SAC) under an adaptive chosen-cipher-text attack is done by Zhou *et al.* (2009). It is observed that SAC is not suitable for applications where the attacker has access to the decoder. The research done by Zhou *et al.* (2009) also presents an improved version of SAC to jointly enhance the security and the performance. Pande *et al.* (2010) have proposed a Joint Video Compression and Encryption (JVCE) framework using Binary Arithmetic Coding (BAC) by Pande *et al.* (2010). The proposed Chaotic Binary Arithmetic Coding (CBAC) scheme uses an interpretation of arithmetic coding using chaotic camps. The compressed data itself is not encrypted, making it easier to preserve properties of video data for indexing, search, network communications and other operations. The proposed security enhancements by Pande *et al.* (2010) lead to the design of video encryption scheme that are resistant to known attacks. A simultaneous arithmetic coding and encryption scheme utilizing chaotic maps has been proposed by Wong *et al.* (2010). It performs better than the traditional arithmetic coding schemes. This is because both the position and direction of line segments in the piecewise linear chaotic map are controlled using a secret key.

A multiple chaotic system for video compression and encryption is proposed by Qian *et al.* (2008). The proposed approach by Qian *et al.* (2008) not only encrypts after compressing video streams but also during the processing of compression frame by frame. The proposed scheme takes advantage of two virtues, namely, high speed of partial encryption and overall security of block encryption without sacrificing overall security and speed. A modified chaos-based joint compression and encryption scheme is proposed by Chen *et al.* (2011). The lookup table used for encryption is dynamically updated during searching. This in turn, reduces the number of chaotic map iterations wasted for visiting irrelevant symbols. A secured arithmetic coding with error detection capability is proposed by Sinaie and Vakili (2010). Katti *et al.* (2011) have analyzed security issues of RAC and observed that RAC is not secured even in the presence of only an eavesdropper and definitely not secured against chosen-plaintext or chosen-ciphertext attacks. They have concluded that making modifications to entropy coding schemes must be done with extreme care (Katti *et al*., 2011). Randomized Matrix Arithmetic Coding (RMAC) is proposed by Kavitha *et al*. (2011) to overcome chosen-ciphertext attack. In this scheme, a randomized matrix is formed by random key based on user profile. Thus, the attacker is unable to guess the cipher or key since, plaintexts are encrypted with symbols in the matrix. RMAC increases the security of arithmetic coding by preventing attacks such as Chosen-Ciphertext Attack type-2 (CCA2) attack and third party attacks.

MATERIALS AND METHODS

Arithmetic encoding procedure is based on the classical recursive probability interval partition known as Elias coding; at each iteration the interval is split in two sub-intervals. In the partition of the probability interval, an AC decides in advance whether the interval related either to the Least Probable Symbol (LPS) or to the Most Probable Symbol (MPS) comes first. This decision is agreed upon between encoder and decoder once and for all. AC is very sensitive to errors and tends to propagate throughout the decoded block. A single erroneous decoding step will make the decoded data completely useless. Unlike Huffman coding (Huffman, 1952) which tends to recover after a certain number of erroneously decoded symbols, AC exhibits very poor resynchronization capabilities (Moo and Wu, 1999). In Arithmetic Coding System when errors are focused which does not know the decoding key, it will not be able to properly decode and render the multimedia content (Grangetto *et al*., 2006). On the other hand, the Randomized Arithmetic Coding (RAC) is based on a random organization of encoding intervals. Only a synchronized decoder is able to interpret the encoded sequence correctly. As the encoding is done on a bit by bit basis and an interval partition is associated with each bit for each bit an independent decision is made as to whether the LPS or MPS subinterval comes first.

Figure 1a and b illustrates the operation of the traditional arithmetic coding and randomized arithmetic coding techniques. In traditional arithmetic coding, the message is split into two intervals, represented by P_0 and P_1. It encodes the input string 001 by selecting a binary number contained in the interval, I, whereas, RAC encodes the same string into different interval I'. In RAC, the interval order for the second is swapped (Grangetto *et al*., 2006). Figure 1c shows the interval order of Enhanced Randomized Arithmetic Coding (ERAC), in which the second bit is started from the starting point of second interval position of first bit. The final interval of ERAC can be represented as I'' which is different from intervals of arithmetic coding and randomized arithmetic coding. ERAC encodes input string by selecting a binary number available in the interval I''. The interval order of the given example is started from LPS first and MPS second in first bits of block. The interval order of second bits of block is started from the end position of first interval. Here, LPS comes first and MPS comes second and the third bit of block's interval was swapped. Then, it starts from the end position of second bits of block's first interval. This third bits of block's first interval that is LPS is defined as I'' and it is a binary number. This change could be done by applying a secret key on RAC encoder which will always keep the LPS as first.

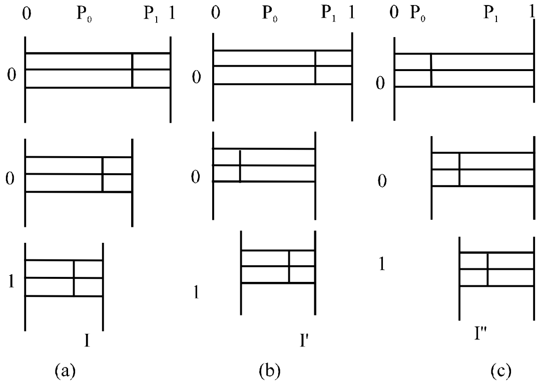


Fig. 1: Various arithmetic coding schemes; a) Traditional Arithmetic Coding; b) Randomized Arithmetic Coding (RAC) and c) Enhanced (RAC)

There are two modules in the proposed design of ERAC, namely, RAC encoder and Key Generator (KG). The input video data is given as symbol by symbol to the RAC encoder for each bit stream. The RAC encoder decides encoding intervals based on a key stream K_1 which is generated by the Key Generator (KG). The output of RAC encoder is bit-wise-XORed with another key stream K_2 which is also generated by the same Key Generator (KG). After the XOR operation, all encoded symbols are combined together to generate the output stream cipher. In Fig. 2, PV and CV represent the input plain video and the output cyber video. The character C denotes the compressed video and E denotes the as another encryption key and should be known by the receiver for decrypting the video data. Therefore, the seed must be unique and confidential. The computational steps involved in the proposed ERAC are given. The decryption

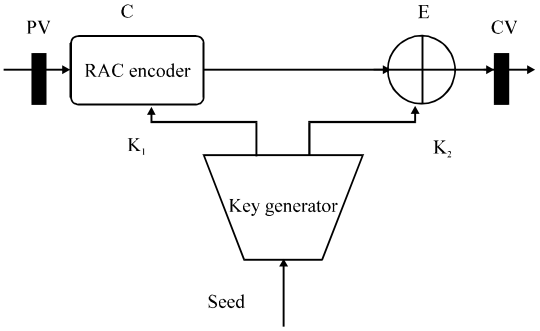


Fig. 2: Components of enhanced randomized arithmetic coder

process is just the reverse of the steps described below. The number of pixels in each block of video stream is N. The keys K_1 and K_2 are of 128 bits and the block size considered is 64 bits.

```
Step 1: Input the video data and the seed S to the RAC encoder
Step 2: Generate two random keys $K_1$ and $K_2$ based on the Seed
Step 3: for I = 0 to N-1
        a  Generate a random number X
        b  Consider the ith bit in input video stream, $b_i$
        if ($X_i$ = = 0)
                then select the order [MPS, LPS]
                for encoded bit $b_i$
        else
                select the order [LPS, MPS] for
                encoded bit $b_i$
Step 4: Construct encoded bit stream $E = E_1E_2E_3 \ldots E_N$ which is output by RAC encoder
Step 5: for I = 1 to N  do  $R_i = E_i \oplus K_2$
Step 6: Output stream cipher $R = R_1R_2 \ldots R_N$.
```

RESULTS AND DISCUSSION

The proposed solution is implemented using C# and tested on various files with, different sizes and formats such as .flv, .avi, .asf, .mpg, etc. The proposed ERAC technique gives better compression ratio and takes less execution time when compared to the randomized arithmetic coding (Grangetto *et al*., 2006). It is observed that the compression ratio achieved depends on the file format and size.

Figure 3 shows a single frame of a sample input video data before and after applying the proposed joint compression and encryption technique, Enhanced Randomized Arithmetic Coding (ERAC). The input video considered in Fig. 3 is of 32574 kbytes. Initially, the given video data is split into frames. Then, the frames are compressed and encrypted using the proposed ERAC technique.

Finally, all encoded frames are combined together to generate the encrypted output video. Table 1 compares the results obtained using the proposed Enhanced Randomized Arithmetic Coding (ERAC) and the existing Randomized Arithmetic Coding (RAC) based on execution time. It is observed that the proposed ERAC performs better than the existing RAC scheme for almost all inputs.

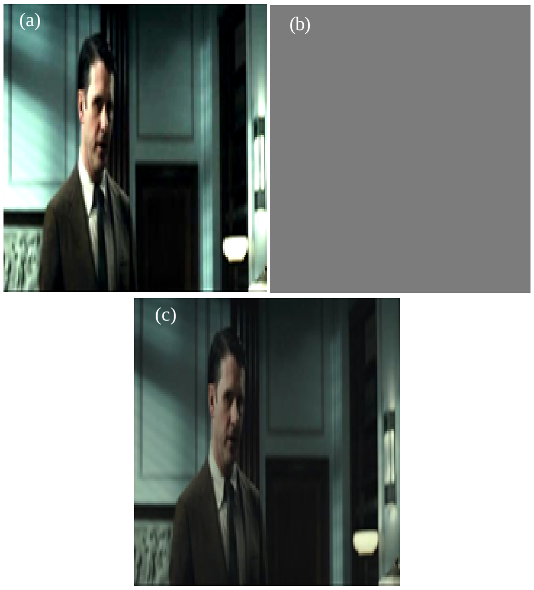


Fig. 3: Video frame before and after applying ERAC; a) Original input video file; b) Compressed and encrypted video and c) Decrypted output video

Table 1: ERAC vs. RAC: comparison based on execution time

		Execution time (msec)	
File name	File size (kbytes)	RAC	ERAC
house.flv	1740.00	9230.75	5807.21
koo.flv	7854.00	39878.00	32897.92
ka.flv	8391.00	43209.25	35453.75
g1.mpg	32574.00	180371.40	154326.25
de.mpg	50930.00	317358.00	286518.50
roboo.mpg	69492.00	408261.00	370423.00
g3.avi	89875.00	472754.00	438321.35
g3.asf	95375.00	487325.50	448927.00
Average	44528.88	244798.50	221584.40

Figure 4 shows the snapshot of video selection (g1.mpg) for joint compression and encryption using ERAC. The snapshot in Fig. 5 shows details such as execution time and file size before and after encryption. The size of the input video is 32574 kbytes and the encoded file size is 28313 kbytes. The total time taken for joint compression and encryption is 154326 msec.

Figure 6 and 7 show the CPU utilization and the memory utilization of proposed Enhanced Randomized Arithmetic Coding (ERAC) and the existing Randomized Arithmetic Coding (RAC). The file house.flv takes 5807.21 msec in ERAC and 9230.75 msec in RAC for

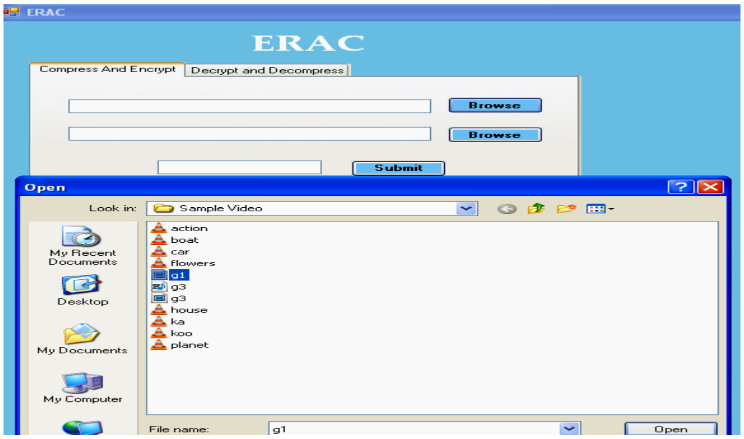
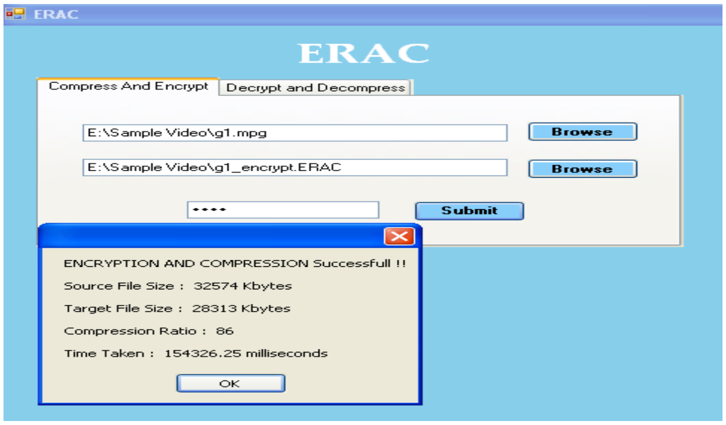


Fig. 4: Video selection for joint compression and encryption in ERAC



E:\Sample Video\g1.mpg

E:\Sample Video\g1_encrypt.ERAC

••••

ENCRYPTION AND COMPRESSION Successfull !!

Source File Size : 32574 Kbytes

Target File Size : 28313 Kbytes

Compression Ratio : 86

Time Taken : 154326.25 milliseconds

Fig. 5: Encoded file size and time

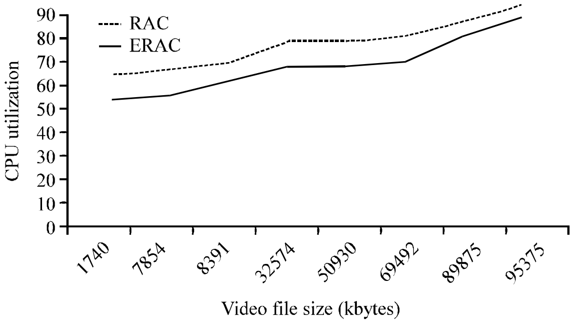


Fig. 6: CPU utilization: RAC vs. ERAC

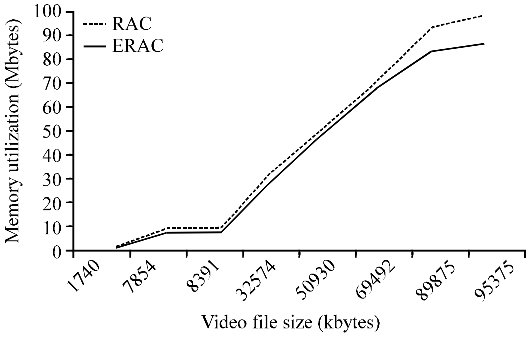


Fig. 7: Memory utilization: RAC vs. ERAC

compression and encryption. The CPU utilization ratio of this file is 54% in ERAC and 65% in RAC. The memory utilization is 1.694 Mbytes in ERAC and 1.835 Mbytes in RAC.

Table 2 tabulates the results of execution efficiency for various files using ERAC. It is known that AC and RAC techniques are prone to ciphertext-only, chosen-plain text and known plaintext attacks (Cleary *et al*., 1999; Grangetto *et al*., 2006; Zhou *et al*., 2009). In this research, the proposed ERAC technique tries to mitigate the above attacks. Researchers ciphertext-only attack or known-cipher-text attack is a type of attack in which the

Table 2: Execution efficiency of ERAC algorithm

				Execution time (msec)	
File name	File size (kB)	Compressed and encrypted	Compression (%)	Compression and encryption	Decryption and decompression
house.flv	1740	1477	84	5807	6420
Koo.flv	7854	6851	87	32897	37756
ka.flv	8391	7404	88	35453	39825
g1.mpg	32574	28313	86	154326	181085
de.mpg	50930	45162	88	286518	312846
roboo.mpg	69492	62093	89	370423	435792
g3.avi	89875	80094	89	438321	472574
g3.asf	95375	82181	86	448927	479318
Average	44528	39196	87	221584	245702

adversary has access to only a limited set of cipher texts. The attack is successful if the corresponding plaintext can be deduced or an educated guess can be made to identify the key with the help of the known set of cipher texts. The proposed ERAC scheme in this study encrypts every symbol of the data by using an XOR operation with a randomly generated key thus making the output stream unrecognized to the attacker. Chosen-plaintext attack is a type of attack in which the attacker has the potential to choose some random plaintexts to be encrypted and obtain the corresponding ciphertexts. The goals of this attack are to gain further information which reduces the security of the encryption scheme and reveal the scheme's secret key with the help of the chosen-plaintext. ERAC mitigates this attack by generating the keys randomly with a given seed which is known only to the sender and the receiver. Thus, the attacker is unable to presume the key since, the key is separated from the cipher text.

Known-plaintext attack is a type of attack in which the attacker has access to samples of both plaintexts and cipher texts. By keenly studying these plaintext-ciphertext pair, the attacker is able to recognize the pattern of encrypted data. In ERAC, the output cipher video is both compressed and encrypted using RAC and XOR operation. Therefore, the probability of determining the pattern by the attacker is negligible.

CONCLUSION

In this research, a joint compression and encryption technique called, Enhanced Randomized Arithmetic Coding (ERAC) is proposed. The proposed technique mitigates security threats such as ciphertext-only, chosen-plaintext and known-plaintext attacks by employing a two step process, namely, modified randomized arithmetic coding and an XOR-operation. Experimental results show that the proposed scheme gives better results in terms of execution time, memory utilization and compression ratio without compromising the quality of video data. Since, this video compression encryption scheme works on bit stream and not on frames, this technique can be extended to other types of data such as image and audio.

REFERENCES

Chen, J., J. Zhou and W. Kwok-Wo, 2011. A modified chaos-based joint compression and encryption scheme. IEEE Trans. Circuits Syst. II: Exp. Briefs, 58: 110-114.

Cleary, J.G., S.A. Irvine and I. Rinsma-Melchert, 1999. On the insecurity of arithmetic coding. Elsevier Comput. Secur., 14: 167-180.

Grangetto, M., A. Grosso and E. Magli, 2004. Selective encryption of jpeg 2000 images by means of randomized arithmetic coding. Proceedings of the International Conference on Multimedia Signal Processing. September 29-October 1, 2004, New York, USA., pp: 347-350.

Grangetto, M., E. Magli and G. Olmo, 2006. Multimedia selective encryption by means of randomized arithmetic coding. Trans. Multimedia, 8: 905-917.

Howard, P.G. and J.S. Vitter, 1994. Arithmetic coding for data compression. Proc. IEEE., 82: 857-865.

Huffman, D.A., 1952. A method of construction of minimum-redundancy codes. Proc. Instit. Radio Eng., 40: 1098-1101.

Katti, R.S., S.K. Srinivasan and A. Vosoughi, 2011. On the security of randomized arithmetic codes against ciphertext-only attacks. Trans. Inform. Foren. Secur., 6: 19-27.

Kavitha, V., S. Balaji and R. Jeeva, 2011. RMAC: A new encryption scheme for arithmetic coding to evade cca attacks. Proceedings of the 3rd International Conference on Advanced Computing, Jan 18-20, 2011, Harbin Institute of Technology, Harbin, China, pp: 175-180.

Kim, H., J. Wen and J.D. Villasenor, 2007. Secure arithmetic coding. Trans. Signal Proces., 55: 2263-2272.

Moffat, A., R.M. Neal and I.H. Witten, 1995. Arithmetic coding revisited. ACM Trans. Inform. Syst., 16: 256-294.

Moo, P.W. and X. Wu, 1999. Resynchronization properties of arithmetic coding. Proceedings of the International Conference on Data Compression, March 29-31, 1999, Snowbird, UT.

Pande, A., J. Zambreno and P. Mohapatra, 2010. Joint video compression and encryption using arithmetic coding and chaos. Proceedings of the International Conference on Internet Multimedia Systems Architecture and Application. December 15-17, 2010, Bangalore, pp: 1-6.

Qian, Q., Z. Chen and Z. Yuan, 2008. Video compression and encryption based-on multiple chaotic system. Proceedings of the 3rd International Conference on Innovative Computing Information, June 18-20, 2008, Dalian, Liaoning, pp: 561-561.

Sinaie, M. and V.T. Vakili, 2010. Secure arithmetic coding with error detection capability. EURASIP J. Inform. Secur., 10.1155/2010/621521.

Socek, D., S. Magliveras, D. Culibrk, O. Marques, H. Kalva and B. Furht, 2007. Digital video encryption algorithms based on correlation-preserving permutations. EURASIP J. Inform. Secur. 10.1155/2007/52965.

Wen, J., H. Kim and J.D. Villasenor, 2006. Binary arithmetic coding with key based interval splitting. Sig. Process Lett., 13: 69-72.

Witten, I.H., R.M. Neal and J.G. Cleary, 1987. Arithmetic coding for data compression. ACM Commun., 30: 520-540.

Wong, K.W., Q. Lin and J. Chen, 2010. Simultaneous arithmetic coding and encryption using chaotic maps. Trans. Circuits Syst. II., 57: 146-150.

Xie, D. and C.C.J. Kuo, 2007. Multimedia encryption using joint randomized entropy coding and rotation in partitioned bitstream. EURASIP J. Inform. Secur., 10.1155/2007/35262.

Zhou, J., O.C. Au and P.H.W. Wong, 2009. Adaptive chosen-ciphertext attack on secure arithmetic coding. Trans. Signal Process., 57: 1825-1838.