Journal of Computer Science 8 (9): 1541-1546, 2012
ISSN 1549-3636

A Block Cipher Algorithm for Multimedia Content Protection with Random Substitution using Binary Tree Traversal

P. Vidhya Saraswathi and M. Venkatesulu
Department of Computer Applications, Kalsalingam University, Krishnankoil, India

Abstract: Many people consume multimedia content (images, music, movie) on portable devices like DVD player, MP3 player, Portable Multimedia Player and also through Internet. **Problem statement:** The conventional algorithms such as DES and AES cannot be used directly in multimedia data, since multimedia data are repeatedly have high redundancy, large-volumes and require real-time operations, such as displaying, cutting, copying, bit-rate conversion and so forth. A block cipher is usually used to encrypt multimedia content because of its reasonable security and performance**. Approach:** In this study, we introduce a naive approach of efficient multimedia content encryption scheme which uses a block of bits rather than bytes or pixels. The proposed block cipher encrypts any type of compressed multimedia content by random substitution using binary tree traversal, row shifting and column shifting. **Results:** Experimental results show that the new algorithm has better performance than DES algorithm, encrypting multimedia content by dividing the plaintext by blocks. **Conclusion:** The proposed algorithm is implemented for all types of multimedia files like audio, video, images and text data and this algorithm can be used to multimedia data during transmisson through Internet or through any communication channels.

Key words: Block cipher, multimedia, encryption, decryption

INTRODUCTION

In today‘s information age, data transmission plays an important role which is contributed to the growth of technologies. Electronic security is increasingly involved in making communications more prevalent. Therefore, a mechanism is needed to assure the security and privacy of information that is sent over the electronic communications media is in need. Whether the communications media is wired or wireless, both can not be protected from unauthorized reception or interception of transmission. While modem cryptography is a vast and complicated field, the basics are easy to understand. In recent years, more and more businesses make use of communication networks, share potential information and therefore sensitive data is located in communications network transmissions that are connected all over the world. This commitment to data communication has increased the vulnerability of organization assets. Computer fraud is becoming one of the most popular crimes in our days.

Cryptography is necessary when communicating over any untrusted medium, which includes just about any particularly, the Internet. Within the context of any application-to-application communication, there are some specific security requirements, including:

- Authentication: The process of proving one‘s identity
- Privacy/confidentiality: Ensuring that no one can read the message except the intended receiver
- Integrity: Assuring the receiver that the received message has not been altered in any way from the original
- Non-repudiation: A mechanism to prove that the sender really sent this message

Digital Rights Management becomes important, which controls digital content usage under wireless environment. In a typical DRM model, a block cipher is usually used to encrypt multimedia content because of its reasonable security and performance. It is because users want long playtime and quick responsiveness with random access.

Multimedia content protection: While using the multimedia content through standby devices and through Internet, the end user wants a quick response for playing multimedia data.In general user feels uncomfortable if the response time exceeds one second. Playing the multimedia data after full decryption may not satisfy user’s requirement due to large size of data. So alternating with decrypting and playing data may be a good method to reduce sensory activation time.

Content protection by symmetric cipher: In cryptography symmetric and asymmetric cipher are used to prevent unauthorized access to multimedia content and illegal distribution. A symmetric key cipher uses the same key for data encryption and decryption and requires two communication parties share the key. The encryption

Corresponding Author: P. Vidhya Saraswathi, Department of Computer Applications, Kalsalingam University, Krishnankoil, India

speed of symmetric key cipher is faster than that of asymmetric key cipher. A block cipher takes fixed-length groups of bits termed blocks from plain text as input and performs permutation and substitution (Schenier, 1996). Finally, same length of block is generated as a cipher text. In CBC, encryption mode of block cipher encrypts each plaintext block with an adjacent cipher text block and key. Therefore, it can decrypt any specified block immediately with key because all blocks are cipher text.

The selective encryption: Multimedia data have different characteristics from text data. It is not necessary to encrypt data completely for protecting a huge multimedia file (Cheng and Li, 2000). In the area of multimedia security, "selective encryption" is devised to protect multimedia content and fulfill the security requirements for a particular multimedia application. Selective encryption is the technique of encrypting some parts of multimedia content while leaving others unencrypted. It may be a good alternative to full encryption since it can cause significant loss of quality during playing. Some multimedia applications such as TV broadcasting require much lower level security. In selective encryption, it is an important issue to determine which parts of data to be encrypted. Possible approaches are to encrypt some important parts of content; to divide content into fragments and then encrypt every N^{th} fragment; or to encrypt randomly chosen parts. Only encrypting some important part can show performance improvement. However, there are no general algorithms to select important parts of content. On the other hand, encrypting every N^{th} fragment of content is practically useless.

Related work: Many encryption algorithms are developed for securing images itself. By applying the principles of cryptography the images can be considered as data blocks or streams. Another method of image encryption is implementing scrambling algorithms for encrypting images by decomposing the original image into its binary bit planes. Zhou *et al*. (2009) proposed an image encryption algorithm by performing XOR operation with key image, inverting the components of bit planes and generate the encrypted image by selected scrambling method. Xiao and Xia (2008) proposed an image encryption algorithm in which the position of images are shuffled and states of hyper chaos are used to change the grey scale of the shuffled image. Amin *et al*. (2010) proposed an image encryption algorithm which encrypts 256 bits plain image to 256 bits cipher image using cryptographic primitive operations and non linear transformations.

Yoon and Kim (2010) proposed a new image encryption algorithm using a large pseudorandom permutation which is combinatorially generated from small permutation matrices based on chaotic maps. Tong and Cui (2008) proposed a new encrypting image scheme using the new compound chaotic function by choosing one of the two one-dimensional chaotic functions randomly. Zhi-Liang *et al*. (2011) proposed an image cryptosystem employing the Arnold cat map for bit-level permutation and the logistic map for diffusion. (Ali *et al*., 2007) proposed a block-based transformation algorithm based on the combination of image transformation and a well known encryption and decryption algorithm called Blowfish.

Video encryption algorithms based on secret key and public key methods are formulated and public key cryptography is not applicable since the operations require large amount of time which is not suitable for video conferencing (Bojnordi *et al*., 2005). Video encryption algorithms can be classified as follows: Naive algorithm, selective algorithm, Zigzag algorithm, RC4 and AES. The idea of naive encryption is to encrypt video streams as byte by byte. Naive algorithm encrypts every byte in the whole video stream and these algorithms guarantee the most security level.

However, it is not an applicable solution if the size of the data is large. In selective algorithm, four levels of selective algorithms are suggested. These four levels are encrypting all headers, encrypting all headers and I (initial) frames, encrypting all I frames and all I blocks in P and B frames and finally encrypting all frames as in Naive algorithm to guarantee the highest security. The idea of ZIG-ZAG algorithm is basically encrypting the video streams before compressing them. Explicitly, when mapping the 8×8 block to a I×64 vector each time in the same order. We can use a random permutation to map this transformation of the 8×8 block to the I×64 vector. Therefore, the concept of the encryption key does not exist in the ZIG-ZAG permutation algorithms. Once the permutation list is known, the algorithm will not be secure any longer.

Shi and Bhargava (1998) proposed a new video encryption algorithm called VEA depends on dividing the video streams into chunks. These chunks are separated into two different lists (odd and even lists). Applying encryption algorithm like DES to the even list and the final cipher is concatenation of output of encryption algorithm XOR with the odd list streams. RC4 is stream cipher structure in which it encrypts plain text one byte at a time with variable length key size from 1to 256 bytes (8-2048). RC4 is a symmetric encryption algorithm in which the same key is used for encryption and decryption. The algorithm is based on the use of random permutation. RC4 is the most widely used stream cipher used in the SSL/TLS (Secure Socket Layer/Transport Layer Security) standards that have been defined for communication between web browsers and servers in which it encrypts plain text one byte at a time with variable length key size from 1-256 bytes.

MATERIALS AND METHODS

Proposed algorithm: The objective of our scheme is both to reduce the computational requirements compared to encrypting a whole file with only a block cipher and to strengthen security comparatively as that of selective encryption. In our proposed algorithm the block of multimedia content is represented as binary tree in the initial step and matrix format in successive steps for row shifting and column shifting.

Encryption algorithm: The proposed algorithm encrypts the plaintext block by block and each block contains 2^{2n} bits. As the first step of the encryption process, each 22n bit plain-block is represented as a complete binary tree.

Step 1: Arrange the block of bits of size 2^{2n} as complete binary tree. The successive bits of the plaintext reside in each level and the construction of complete binary tree continues until for all bits of the plaintext. Denote the plaintext f = B (l, x), l denotes each level of the binary tree and x would be the position of the node according to the permutation position, the MSB is at root node and the consecutive bits are added as left and right child at each level and the LSB is attached as leaf node and this node can be of left or right child of any node in the previous level of leaf nodes.

Step 2: A random permutation P is generated by key based permutation algorithm, so that P= $\{P_1,P_2,P_3..P_n\}$ is the subset of {1, 2…n}.

Step 3: Apply randomized substitution by choosing the node x at the position P_i of the binary tree.

Step 4: Let z denote the sum of bit values at all nodes, starting from root node, traversing all levels until the node x with the sum of all nodes of the sub tree rooted at x and the value:

$$z = \sum_{j=0}^{l} B(j,x) + \sum_{\substack{k,\text{all}\\ \text{nodes}\\ \text{in sub tee}\\ \text{rooted at x}}} B(k,x) \qquad (1)$$

where, B(j,x) in Eq. 1 denotes the value at a node on level j along the path from the root node to x and B(k,x) denotes the value at a node k in the sub tree rooted at x:

Step 5: If z=0, the value at node *x* is replaced by 0, otherwise the value at node *x* is replaced by 1

Step 6: Repeat the process for all nodes at positions $P_1,P_2,P_3...P_n$ and the resultant bits are termed as pseudo cipher text C_1

Step 7: Arrange C_1 into log√n X log√n matrix A and assign the permutation position $P_1….P_n$ to each element $A_{ij.}$ Find the sum of permutation positions for each row. Let M be the count of 1's in each row. If the permutation sum is odd then perform row wise left shifting M times otherwise perform row wise right shifting M times

Step 8: Perform column wise downward shifting M times if the sum of permutation along column wise is even, otherwise perform column wise upward shifting M times. The resultant bits are termed as cipher text C

Decryption algorithm: The cipher text C is given as input for the decryption:

Step 1: Reverse the permutation P as $P_n,P_{n-1}…P_1$

Step 2: Construct the binary tree for each block of the cipher text as the same way of encryption

Step 3: Apply randomized substitution by choosing the node *x* at the position P_i of the binary tree

Step 4: Let z' in Eq. 2 denote the sum of bit values at all nodes, starting from root node, traversing all levels until the node x with the sum of all nodes of the sub tree rooted at x:

$$z' = \sum_{j=0}^{l} B(j,x) + \sum_{\substack{k,\text{all}\\ \text{nodes}\\ \text{in sub tee}\\ \text{rooted at x}}} B(k,x) \qquad (2)$$

Step 5: If z'=0, the value at *x* is replaced by 0, otherwise the value at *x* is replaced by 1.

Step 6: Repeat the process for all nodes at positions $P_n,P_{n-1}P_{n-2}...P_1$ and the resultant bits are termed as pseudo plain text

Step 7: Arrange C_1 into log√n X log√n matrix A and assign the permutation position $P_1….P_n$ to each element $A_{ij.}$ Find the sum of permutation positions for each row. Let M be the count of 1's in each row. If the permutation sum is odd then perform row wise right shifting for M times otherwise perform row wise left shifting for M times

Step 8: Perform column wise upward shifting M times if the sum of permutation along column wise is even, otherwise perform column wise downward shifting M times. The plaintext bits are retained after decryption

Properties of the proposed algorithm:

Property 1. The algorithm takes O (n), both for encryption and decryption.

Proof: The creation of binary tree takes O (n) time. The generation of pseudo random permutation takes O (1) time. Each substitution takes o (1) time. Since there are n elements to be replaced, substitution takes O (n) time and it is shown in Eq. 3:

$$\sum_{l=1}^{k}(1+2^{k-1}) = \frac{k(k+1)}{2} + 2^{k}\left(1-\frac{1}{2^{k-1}}\right) = 0(n) \quad (3)$$

Property 2: The algorithm correctly decrypts the cipher text C into the original plain text.

Proof: A node x in binary tree is taken and let x occupy the position P_i in forward substitution. Let z be equal to the sum of all values at all the nodes starting from the root node up to x plus the values at all nodes of the sub tree rooted at x.

Case 1: If z is even the value a = 0 at node *x*, if already a = 0 the result b = 0, since z-a is even number, if a = 1 at node x the result b = 0, z-a becomes odd. In decryption the value at b=0 at node *x* is retained as b = 0 = a, because z-b, z-a = even, if a = 1 then a is changed to b=0,because z-b = z-0 = (z-a)+a is even. Since (z-a) is odd, b is changed to a = 1.

Case 2: If z is odd, the value a=0 at node x is changed to b=1, therfore z= z-a = (z-1)+1 = (z-a)+a, if and only (z-a) is odd.

In decryption the value b = 1 at node *x* is changed to a=0,therefore z = (z-a)+a = (z-a)+1=even,b=1 is changed to a = 0.If a = 1 then a is returned as b=1,z = (z-a)+a is odd,(z-a) is even, z = (z-a)+1 is odd, b = 1 is returned as a = 1.

RESULTS

The proposed algorithm is experimented for all types of multimedia files (images, music and videos).The multimedia content of any type is converted into binary format and applied to encryption. The encrypted image of lena image by Blowfish algorithm and proposed algorithm is given in Fig. 1a-c. The results for music and video files show that the both encryption and decryption time for music and video files are lesser than their play time, so playing of both the files are started parallel along with decryption.

The Table 1 and 2 show the experimental results of encrypion and deryption time for image files, music files and video files.

The results for music and video files show that the both encryption and decryption time for music and video files are lesser than their play time,so playing of both the files are started parallel along with decryption.

Table.1: Comparison of Encryption time of proposed algorithm with DES algorithm

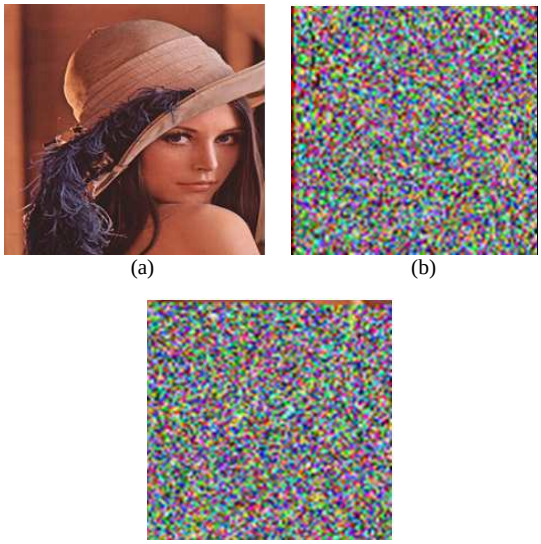
File type	File size	Encryption time	
		DES	Proposed algorithm
Image(.jpg)	4.3 MB	4.3 min	9.56 sec
Audio(mp3)	4.7 MB (play time:6 min)	4.4 min	7.44 sec
	76MB(play time:3 min)	18.48 min	1.56 min
Video(mp4)	255MB (play time:15 min)	45.37 min	7.13 min

DISCUSSION

Security analysis: This section addresses the security of the proposed encryption technique and analysis of experimental results.

The pseudo random permutation which is generated by the key value has no influence on the plaintext recovered from the decryption process. It is because the key is only used to determine the pseudo random permutation and never used to change the value of any other bit in the plaintext.

Differential attack: To test the influence of one-pixel change on the whole encrpted image, two common measures NPCR and UACI are used. The Number of Pixels Change Rate (NPCR) measures the different pixel numbers between two images and UACI(Unified Average Changing Intensity) measures the average intensity of differences between the plainimage and the cipherimage. For the calculation of NPCR and UACI, we have taken two encrypted images E_1 and E_2 and assume their corresponding plainimages have only one-pixel difference.



(a) (b) (c)

Fig. 1: (a) Plainimage), (b) Encrypted image by Blowfish algorithm, (c) Encrypted image by Proposed algorithm

Table 2: Comparison of decryption time of proposed algorithm with DES algorithm

File type	File size	Decryption time	
		DES	Proposed algorithm
Image (.jpg)	4.3 MB 4.7 MB	4.5 min	11.14 sec
Audio (mp3)	(play time :6 min)	4.8 min	9.98 sec
	76MB (play time: 3min)	25.48 min	2.43 min
Video(mp4)	255MB(play time:15 min)	52.37 min	11.28 min

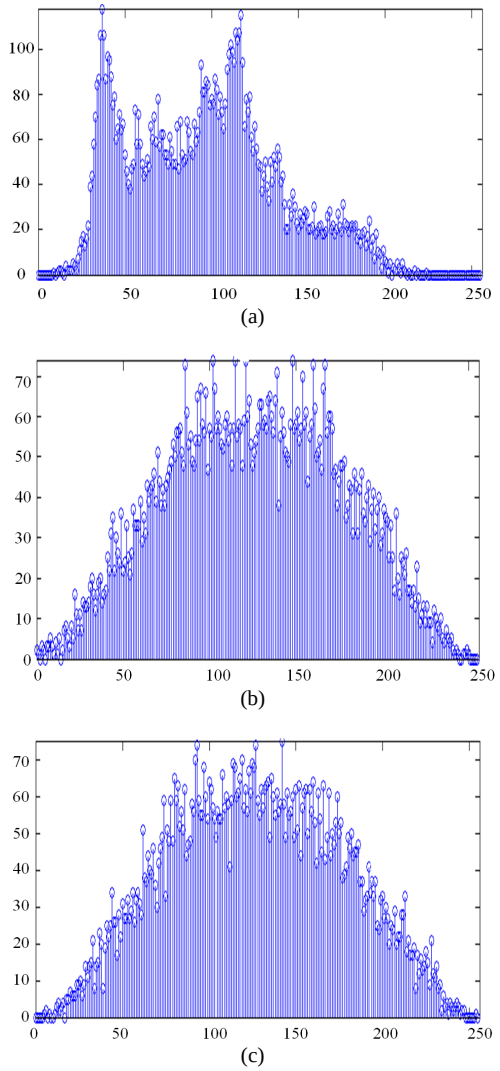


Fig. 2: (a) Histgram of Plain image,(b) Histogram Encrypted image by Blow fish algorithm,(c) Histogram of Encrypted image by proposed algorithm

Let W and H are the width and height of image and the gray-scale values of the pixels at grid (i,j) of E_1 and E_2 are labeled as $E_1(i,j)$ and $E_2(i,j)$ respectively. Define a bipolar array, D, with the same size as images E_1 and E_2. Then D(i,j) is related to $E_1(i,j)$ and $E_2(i,j)$, if $E_1(i,j) = E_2(i,j)$, then D(i,j) = 1 else D(i,j) = 0. The two measures NPCR and UACI are defined in Eq. 4 and 5 are given below:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W.H} * 100 \quad (4)$$

$$UACI = \frac{1}{W.H} * \frac{\sum_{i,j} |E1(i,j) - E2(i,j)|}{255} * 100 \quad (5)$$

Tests have been performed on the proposed algorithm, taking randomly a pixel of the original image and make a slight change on the gray-scale level of this pixel. The encryption algorithm is performed on the modified original image and the two measures NPC R and UACI are computed. We obtained NPCR = 99.85% and UACI = 33.58%. The results show that a slight change in the original image results in a great change in the encrypted image implies that the proposed algorithm has a good capability to resist the differential attack.

Histogram analysis: An image histogram illustrates how pixels in an image are distributed by graphing the number of pixels at each color intensity level. The Fig. 2 gives the histogram of plain images and encrypted images. We have calculated and analyzed the histograms of the several encrypted images as well as its original images that have widely different content. The histogram of the cipher images from proposed algorithm are shown in Fig 2. The encrypted images bear no statistical resemblance to the plainimage. It is clear that the histogram of the encrypted image is fairly uniform and significantly different from the respective histograms of the original image and hence does not provide any clue to employ any statistical attack on the proposed image encryption procedure.

Correlation analysis: We have also analyzed the correlation between two vertically adjacent pixels, two horizontally adjacent pixels and two diagonally adjacent pixels in plain image and cipher image, respectively. The procedure is done by randomly selecting 100 pairs of two adjacent pixels from an image. Then, the correlation coefficient is calculated using the following formulas in Eq. 6-9:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{Dx} * \sqrt{Dy}} \quad (6)$$

$$E(x) = \frac{1}{n}\sum_{i=1}^{n} xi \quad (7)$$

$$D(x) = \frac{1}{n}\sum_{i=1}^{n} (x_i - E(x))^2 \quad (8)$$

$$cov(x,y) = \frac{1}{n}\sum_{i=1}^{n} (x_i - E(x)(y_i - E(y)) \quad (9)$$

Correlation coefficients of randomly chosen 100 pairs of two adjacent pixels have calculated for plain image, cipher image encrypted by Blowfish algorithm and the cipher image by proposed algorithm. The correlation coefficents of plain image and encrypted images are given in Table 3.

Table 3: Correlation coefficients of images

Direction	Plain image	Encrypted image (Blowfish)	Encrypted image (proposed)
Horizontal	0.9816	0.0824	0.01776
Vertical	0.9858	0.0898	0.04912
Diagonal	0.9712	0.0548	0.00348

Cipher text only attack: In the cipher text-only attack, the attacker has to find the original values from the encrypted values. According to our algorithm, for each cycle of the encryption the node value of the permutated position of the plaintext may or may not be changed related to the summation of node values along the path and the summation of the sub tree of the permutated position. Though the attacker is familiar with summation, based on that, the pseudo random permutation position P_i cannot be extracted since the replacement of bits are not performed for all substitutions.

Known plain text attack: In the known-plaintext attack, unauthorized user has both original plain text and the corresponding encrypted values. If we choose a sufficiently long plaintext sequence $M_1,M_2,...,M_n$ and its corresponding cipher text $C_1C_2\ldots. C_n$, look for a repetition in the cipher text, i.e., $C_{n1} = C_{n2}$ for some integers n1 < n2,but the probabilities of occurring such cipher texts are low, so the attacker would not able to determine the pseudo random permutation sequence and also this sequence is mainly used for the bit position substitution.

Chosen plaintext attack: Suppose that the attacker has a privilege to execute the encryption machinery, he can choose plaintexts and generate their corresponding cipher text to recover the equivalent pseudo random permutation of bit positions to be traversed along the binary tree form of plaintext. Suppose the attacker chooses a plain text with all zeros as input to the encryption machinery and the cipher text would also be zeros not revealing the pseudo random permutation, which is used as key. From the cipher texts generated by chosen sequence of plain texts the number of 0`s and the number of 1's can be found out. But it is difficult to find out the order of 0's and 1's, since it amounts to checking for all n/2! possible permutations.

CONCLUSION

In this study, a new block cipher algorithm for multimedia cryptosystems is proposed. Based on the pseudo random permutation and substitution, using binary tree traversal, this proposed scheme encrypts any compressed multimedia content. While traditional algorithms and some existing chaotic schemes suffer from the poor diffusion operation, slow performance and small key space, our scheme has effective performance speed. The scheme is more secure for differential attacks, known plaintext attack, chosen plain text attack and able to encrypt large data sets with efficient and secure way. So, our algorithm is promising for real-time applications.

REFERENCES

Ali, M., B. Younes and A. Jantan, 2007. Image encryption using block-based transformation algorithm. Int. J. Comput. Sci., 35: 1-9.

Amin, M., O.S. Faragallah and A.A.A. Et-Latif, 2010. A chaotic block cipher algorithm for image crypto systems. Commun. Nonl. Sci. Numer. Simulat., 15: 3484-3497. DOI: 10.1016/j.cnsns.2009.12.025

Bojnordi, M.N., M.R. Hashemi and S.O. Fatemi, 2005. Implementing an efficient encryption block for MPEG video streams. Proceedings of 47th International Symposium, Jun. 8-10, IEEE Xplore Press, Zadar, pp: 127-130. DOI: 10.1109/ELMAR.2005.193659

Cheng, H. and X. Li, 2000. Partial encryption of compressed images and videos. IEEE Trans. Signal Proces.,48: 2349-2451. DOI: 10.1109/78.852023

Schenier, B., 1996. Applied Cryptography. 2nd Edn., John Wiley and Sons, New York.

Shi, C. and B. Bhargava, 1998. An efficient MPEG video encryption algorithm. Proceedings of the 7th IEEE Symposium on Reliable Distributed Systems, Oct. 20-23, IEEE Xplore Press, West Lafayette, IN, pp: 381-386. DOI: 10.1109/RELDIS.1998.740527

Tong, X. and M. Cui, 2008. Image encryption with compound chaotic sequence. Image Vis. Comput., 26: 843-850. DOI: 10.1016/j.imavis.2007.09.005

Xiao, Y. and L. Xia, 2008. A new hyper-chaotic algorithm for image encryption. Proceedings of the 9th International Conference for Young Computer Scientists, Nov. 18-21, IEEE Xplore Press, Hunan, pp: 2814-2818. DOI: 10.1109/ICYCS.2008.411

Yoon, J.W. and H. Kim, 2010. An image encryption scheme with a pseudorandom permutation basedon chaotic maps. Commun. Nonlinear Sci. Numer. Simulat., 15: 3998-4006. DOI: 10.1016/j.cnsns.2010.01.041

Zhi-Liang, Z., W. Zhang, K.W. Wong and H. Yu, 2011. A chaos-based symmetric image encryption scheme using a bit-level permutation. Inform. Sci., 181: 1171-1186. DOI: 10.1016/j.ins.2010.11.009

Zhou, Y., K. Panetta and S. Agaian, 2009. Image encryption using binary key-images. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Oct. 11-14, IEEE Xplore Press, San Anbnio, USA., pp: 4569-4574. DOI: 10.1109/ICSMC.2009.5346780