ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

Secure data communication using image encryption and compression

Y.M.Kamble[1], K.B.Manwade[2]

P.G.Student, Department of CSE, D.Y.Patil College of Engineering, Kolhapur, India[1]

Associate Professor, Department of CSE, Ashokrao Mane College of Engineering, Vathar tarf Vadgaon, India [2]

Abstract: For secure communication, original data may convert into unintelligent format. Encryption technique is used for converting plain data into scrambled message. Since, as a channel provider, for secure communication, may tend to compress the encrypted data due to limited channel resource. As a traditional data processing for privacy usually takes place before encryption or after decryption. This paper focuses on new concept that first encrypt the given original color image then compress the encrypted image using lossy compression technique without any error and without degrading the quality of natural image. The original color image can be encrypted using standard image encryption technique and then perform lossy compression operation on encrypted image. At receiver end user can recover the original content without any error by performing decryption operation to recover original image.

Keywords: cryptography, image encryption, compression, decryption, image recovery

I. INTRODUCTION

In recent years, cryptography domain has attracted considerable research interest. The effective and popular means of privacy protection is to encrypt the ordinary data into unintelligible data, so as traditional data processing for privacy usually takes place before encryption or after decryption[5]. In some cases content owner does not trust the processing of service provider, when to manipulate encrypted data keeping the plain content unrevealed is desired. For instance, when the secret data to be transmitted are encrypted, a channel provider without any knowledge of the cryptographic key may tend to compress the encrypted data due to the limited channel resource. While an encrypted binary image can be compressed with a lossless manner by finding the syndromes of low-density parity-check codes [1], a lossless compression method for encrypted gray image using progressive decomposition and rate-compatible punctured turbo codes is developed in [2]. Now in the proposed system, the lossy compression method is performed on encrypted color image.

In the first phase content owner will perform encryption operation on given color image by symmetric block cipher algorithm. Then perform lossy compression operation on that encrypted image. When receiver will receive an encrypted compressed image, he will recover the original color image without any error by performing decryption algorithm and using decryption key. Section II presents the proposed scheme where as next sections III and IV presents experimental results followed by conclusion.

II. PROPOSED SCHEME

In the proposed scheme the first phase image encryption will be carried out by using blowfish, symmetric, block cipher technique. Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times. The block size is 64 bits, and the key can be any length from 32 bit to 448 bits.

a) Key Generation

Blowfish makes use of a key that range from 32 bit to 448 bits. The key is used to generate eighteen 32 bit sub keys. The key is stored in a K-array K1, K2, K3 …Kj, where $1 \le j \le 14$. The sub keys are stored in P array, P1 to P18. Initialize first P array using the bits of fractional part of the constant π. Thus leftmost 32 bits of the fractional part of π becomes P1 and so on. Perform a bitwise XOR of the P array and K-array reusing words from the K-array as needed. E.g. P1=P1⊕K1, P2=P2⊕K2….P14=P14⊕K14,P15=P15⊕K1…P18=P18⊕ K4.

b) S-Box Design

The proposed system for encryption of image it requires four S-Boxes of size 8 x 32.

We use CAST128 algorithms four fixed size S-Boxes for encryption purpose. From the given 8 bit input data it will search exact entry from s-box containing hexadecimal value. From that hexadecimal value the required 32 bit data will be produced.

c) Encryption

Encryption is done by 16 round Feistel cipher and uses four S-Boxes from CAST-128.

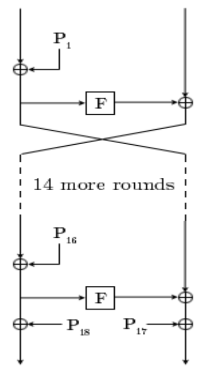

Fig.01 Feistel Structure of Blowfish

The diagram shows the action of Blowfish. Each line represents 32 bits. The algorithm keeps two subkey arrays: the 18-entry P-array and four 256-entry S-boxes. The S-boxes accept 8-bit input and produce 32-bit output. One entry of the P-array is used every round, and after the final round, each half of the data block is XORed with one of the two remaining unused P-entries. The plain text is divided into two 32-bit halves LE_0 and RE_0. After ith round we use variables LE_i and RE_i to refer left and right halve of data. The pseudo code for algorithm is as follows, The input is a 64-bit data element, E. Divide x into two 32-bit halves: LE, RE.
for i=1 to 16 do

$$RE_i = LE_i \oplus P_i;$$

$$LE_i = F[RE_i] \oplus RE_{i-1};$$

$$LE_{17} = RE16 \oplus P_{18};$$

$$RE_{17} = LE16 \oplus P_{17};$$

After the sixteenth round, swap LE and RE again to undo the last swap.
Then, RE = RE XOR P17 and LE = LE XOR P18. Finally, recombine LE and RE to get the cipher text. Decryption is exactly the same as encryption, except that P1, P2... P18 are used in the reverse order.
The resulting cipher text is contained in the two variables LE_{17} and RE_{17}. The diagram shows Blowfish's F-function. The function splits the 32-bit input into four bytes, and uses the bytes as input to the S-boxes. The S-Boxes from CAST-128 contains entries in hexadecimal format. With given 8 bit input to S-Box, the correct entry will be identified and converted to 32 bit binary format.

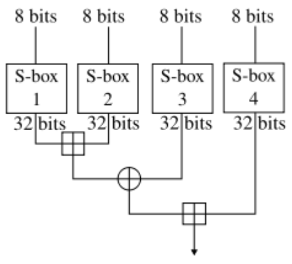

Fig.02 The round function of Blowfish

If we label those bytes a, b, c, and d then the function can be defined as follows,

$$F[a, b, c, d] = ((S_{1,a} + S_{2,b}) \oplus S_{3,c}) + S_{4,d}$$

The advantage of blowfish encryption is that it is one of the strongest algorithms available and the speed of the algorithms and key strength is also very good. Some existing results of blowfish algorithm are as following.
An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

Input Size	DES	3DES	AES	BLOWFISH
20,527	02	07	04	02
36,002	04	13	06	03
45,911	05	17	08	04
59,852	07	23	11	06
69,545	09	26	13	07
137,325	17	51	26	14
158,959	20	60	30	16
166,364	21	62	31	17
191,383	24	72	36	19
232,398	30	87	44	24
Avg Time	14	42	21	11
Bytes/Sec	**7,988**	**2,663**	**5,320**	**10,167**

Table.01 Comparative execution time (in seconds) of encryption algorithms.

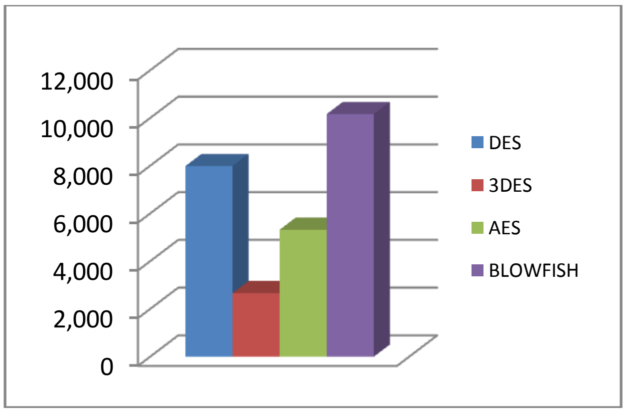

Fig.03 Encryption speed of algorithms bytes per second

d) Compression of Encrypted Image

In the second phase the given encrypted image get compressed by using lossy compression technique. As we have mentioned above, when the secret data to be transmitted are encrypted, a channel provider without any knowledge of the cryptographic key may tend to compress the encrypted data due to the limited channel resource. Hence in the proposed scheme we are using VQ (Vector Quantization) [6, 7]-LBG lossy compression algorithm to compress given encrypted image.

The scheme works as follows:

To compress image vector quantization uses LBG (Linde-Buzo-Gray) algorithm. It is based on minimization of the squared-error distortion measure. The basic idea of VQ is that if a set of representative image vectors, also called codewords, can be designed, they can be then used to represent all the image blocks. LBG algorithm has similar functionality that of K-means clustering algorithm. LBG algorithm for clustering for data compression start with an initial set of centers, make a pass over the image to assign each pixel to its nearest center, compute the centroids of each clusters so obtained and use these centroids as a centers for the next iteration. The final centers will depend on the initial centers that are chosen. The performance of VQ [3, 4] image compression technique depends upon the constructed codebook. To design VQ codebook the Linde-Buzo-Gray (LBG) algorithm is widely used. The performance of the LBG algorithm is highly dependent on initial codebook. To perform compression operation we

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

have used VQ, where the given cipher image is decomposed into non overlapping sub image blocks. These blocks are termed as vectors. Now from all these vectors a set of representative image vectors are selected. The selected image vectors represent the entire set of image blocks. The set of selected representative image vectors are called as a codebook. Each representative image vector from the codebook is called as codeword. Quantization is a process of representing a large set of values with a much smaller set. Two terms mainly used in quantization are Encoder and Decoder. In encoding operation, each code vector is assigned a binary index. Then the input vector from codebook is compared to each code vector from cipher image to find the closest code vector. The index table contains the many to one irreversible mapping of code vectors. The decoder operation takes input as a codeword from codebook and produces output as the estimated value. For this decoder make use of the index table. To inform the decoder about which code vector was found to be the closest to the input vector we transmit or store the binary index of code vector.

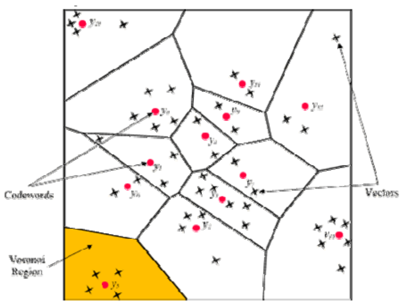

Fig.04 Representative Vectors (Centroid).

LBG Algorithm

- ➢ Determine the number of code vectors N
- ➢ Select N code vectors at random to be the initial codebook
- ➢ Using the Euclidean distance measure cluster the vectors around each code vector
- ➢ Compute the new set of code vectors(codebook)
- ➢ Repeat Steps 2 and 3 until the either the representative code vectors do not change

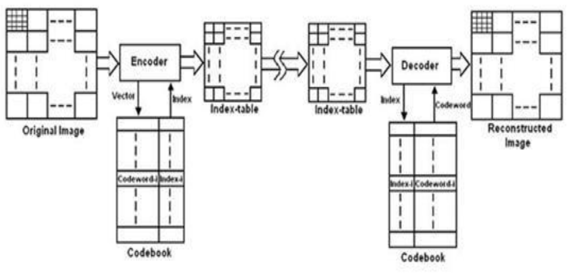

Fig.05 The schematic diagram of VQ encoding-decoding process.

Flowchart for VQ-LBG

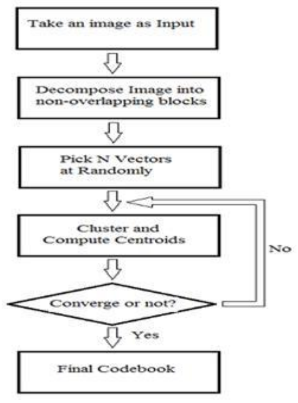
Take an image as Input
Decompose Image into non-overlapping blocks
Pick N Vectors at Randomly
Cluster and Compute Centroids
No
Converge or not?
Yes
Final Codebook

Fig.06 Flowchart of LBG algorithm

The VQ LBG algorithm is an iterative algorithm. Used to generate the color vectors clusters. It work same as K-means clustering algorithm as mentioned above.

III. Experimental Results

To show experimental results we have taken some sample images and performed encryption and compression operation on the given input. While performing compression operation we have tested it for different size of vectors for compression algorithm.

.Image	Original Size in kb	Size after encryption in kb	Image Size after performing compression operation for various vector size in kb			
			64 bit	128 bit	256 bit	512 bit
Sample-1	12.30	12.60	09.62	11.80	11.80	11.80
Sample-2	04.29	12.50	11.80	11.80	11.80	11.80
Sample-3	10.00	10.90	09.97	09.98	09.97	09.90
Sample-4	10.70	10.20	09.40	09.53	09.47	09.46
Sample-5	06.42	11.50	10.70	10.70	10.70	10.70
Sample-6	06.09	10.00	09.21	09.22	09.20	09.19

Table.02 Compression of encrypted image on various vector sizes

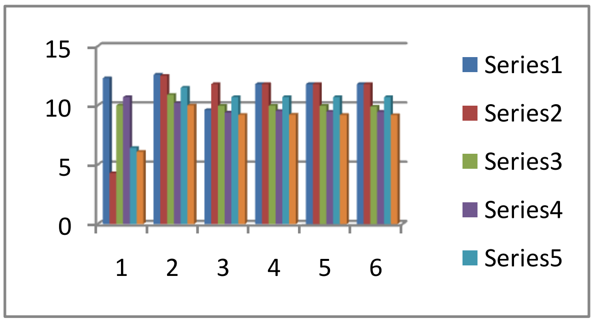

Fig.07 Image compression on different vector size

The graph is drawn as images v/s size in KB. In the graph shown above first group of columns on x-axis shows the original size of the sample images. The second group of columns shows the image size after performing encryption algorithm which increases slightly. Thereafter in each of the columns from group 3 to 6 shows the compression of encrypted image with respect to vector size 64 bit, 128 bit, 256 bit, 512 bit

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

To measure the distortion in decrypted image we have used PSNR as quality metrics. Typical values for the PSNR in lossy image are between 30 and 50 dB. PSNR (Peak signal to noise ratio) can be calculated using following standard formula

$$PSNR=10 \log_{10} ((255*255*255)/MSE$$

Where MSE is mean square error calculated using the square of difference between original image and recovered image divided by image size & by three.
Following table shows the PSNR values of decrypted recovered images. The PSNR values are calculated for different vector size as shown in table and which is approximately 37dB.

Image	PSNR Values for Decrypted Image for Different Vector Size of Compression Algorithm			
	64 bit	128 bit	256 bit	512 bit
Sample-1	37.0183	37.0183	37.0183	37.0183
Sample-2	36.892	44.659	44.650	44.659
Sample-3	37.667	36.667	36.667	36.667
Sample-4	35.920	35.920	35.920	35.920
Sample-5	40.689	40.689	40.689	40.689
Sample-6	40.944	40.944	40.944	40.944

Table.03 PSNR values in dB for decrypted recovered image

IV. CONCLUSION

From the experiments carried out for proposed work it can be concluded that,The symmetric block cipher encryption algorithm, Blowfish, encrypts the given color image and has standard average encryption speed 10,167 bytes/sec.
The lossy compression algorithm, VQ-LBG, compresses the given as shown in result in table 02.The decrypted recovered image maintains the standard image quality in dB as shown in result table 03.

REFERENCES

[1] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data,"IEEE Trans. Signal Process., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.
[2] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of en-crypted grayscale images,"IEEE Trans. Image Process., vol. 19, no. 4, pp. 1097–1102, Apr 2010.
[3] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers, Boston, MA; 1991.
[4] R. M. Gray, "Vector quantization", IEEE Acoustics, Speech, and Signal Processing Magazine, Vol. 1, No. 2, pp. 4-29, April 1984
[5] "SEPARABLE REVERSIBLE DATA HIDING IN ENCRYPTED IMAGE" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 7, NO. 2, APRIL 2012
[6] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers, Boston, MA; 1991.
[7] R. M. Gray, "Vector quantization", IEEE ASSP Magazine, Vol. 1, no. 2, pp. 4-29, April 1984.