Securing Image Transmission Using In- Compression Encryption Technique

Shaimaa A. El-said Eng.sahmed@windowslive.com
Faculty of Engineering / Electronics and Communication Department
Zagazig University
Zagazig,44519, Egypt.

Khalid F. A. Hussein khalid_elgabaly@yahoo.com
Electronics research institute
Microwaves Department
Researches National Institute
Dokki, Egypt

Mohamed M. Fouad fouadzu@hotmail.com
Faculty of Engineering / Electronics and Communication Department
Zagazig University
Zagazig,44519, Egypt.

Abstract

Multimedia is one of the most popular data shared in the Web, and the protection of it via encryption techniques is of vast interest. In this paper, a secure and computationally feasible Algorithm called Optimized Multiple Huffman Tables (OMHT) technique is proposed. OMHT depends on using statistical-model-based compression method to generate different tables from the same data type of images or videos to be encrypted leading to increase compression efficiency and security of the used tables. A systematic study on how to strategically integrate different atomic operations to build a multimedia encryption system is presented. The resulting system can provide superior performance over other techniques by both its generic encryption and its simple adaptation to multimedia in terms of a joint consideration of security, and bitrate overhead. The effectiveness and robustness of this scheme is verified by measuring its security strength and comparing its computational cost against other techniques. The proposed technique guarantees security, and fastness without noticeable increase in encoded image size.

Keywords: Image encryption and compression, optimized multiple Huffman tables, OMHT performance analysis, computational cost analysis

1. INTRODUCTION

With the rapid development of multimedia and network technologies, the security of multimedia becomes more and more important, since multimedia data are transmitted over open networks more and more frequently. Typically, reliable security is necessary to content protection of digital images and videos. Encryption schemes for multimedia data need to be specifically designed to

protect multimedia content and fulfill the security requirements for a particular multimedia application. For example, real-time encryption of an entire video stream using classical ciphers requires heavy computation due to the large amounts of data involved, but many multimedia applications require security on a much lower level, this can be achieved using selective encryption that leaves some perceptual information after encryption.

As an important way of designing a secure video encryption schemes, secret Multiple Huffman Tables (MHT) have been suggested in some designs. The major advantage by using this kind of joint compression-encryption approach is that high compression ratio and high encryption degree can be achieved in one single step, which simplifies the system design and makes it flexible for some advanced multimedia processing [1] in addition to the reduction of time required to perform compression followed by encryption. After re-studies the security of multimedia encryption scheme based on secret Huffman tables, the present cryptanalysis shows presence of drawbacks in MHT technique.

To overcome the drawbacks of MHT technique, a new scheme for more general and efficient secure multimedia transmission, OMHT, is proposed. OMHT depends on using statistical-model-based compression method to generate different tables from a training set has the same data type as images or videos to be encrypted leading to increase compression efficiency and security of the used tables. Using known fixed tables in MHT technique generated by mutation (a method introduced in [1]) for compressing and encrypting images causes degradation in both compression ratio and security. We focus our research attention to enhancing multiple Huffman tables coding techniques. It is a challenging problem to verify joint consideration of security, bitrate overhead, and friendliness to delegate processing. Performance analysis of the newly proposed scheme OMHT shows that it can provide superior performance over both generic encryption and MHT in the security and compression.

This paper is organized as follows: Section 2 shows an overview of multimedia encryption techniques. A new proposed scheme, Optimized Multiple Huffman tables coding technique (OMHT) is described in section 3 with a detailed description for proposed adaptive quantization technique. Section 4 presents a performance analysis of the proposed scheme OMHT technique. The computational cost of the proposed technique is analyzed in section 5. Conclusion is given in section 6.

2. OVERVIEW of MULTIMEDIA ENCRYPTION TECHNIQUES

When dealing with still images, the security is often achieved by using the naïve (traditional) approach to completely encrypt the entire image, traditional encryption, with a standard cipher [2] (DES, AES, IDEA, etc.). As shown in Fig. (1), assuming that the plaintext and the ciphertext are denoted by P and C, respectively, the encryption procedure in a cipher can be described as C = E Ke (P), where Ke is the encryption key and E(·) is the encryption function. Similarly, the decryption procedure is P =DKd (C), where Kd is the decryption key and D(·) is the decryption function When Ke = Kd, the cipher is called a private-key cipher or a symmetric cipher For private-key ciphers, the encryption-decryption key must be transmitted from the sender to the receiver via a separate secret channel. When Ke ≠ Kd, the cipher is called a public-key cipher or an asymmetric cipher. For public-key ciphers, the encryption key Ke is published, and the decryption key Kd is kept private, for which no additional secret channel is needed for key transfer. Ciphering the complete compressed file may result in excessive computational burden and power consumption at the decoder and perhaps even the server/ encoder.

However, there are number of applications for which the naive based encryption and decryption represents a major bottleneck in communication and processing. Some recent works explored a new way of securing the content, named, partial encryption or selective encryption, soft encryption, perceptual encryption, by applying encryption to a subset of a bitstream. The main goal of selective encryption is to reduce the amount of data to encrypt while achieving a required level of security [3].

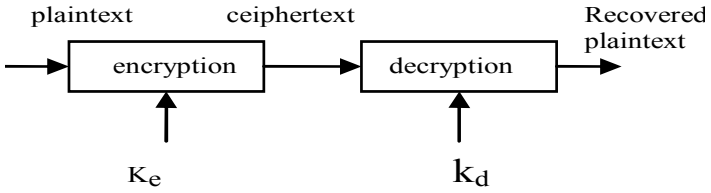


Figure 1: Traditional Encryption Techniques

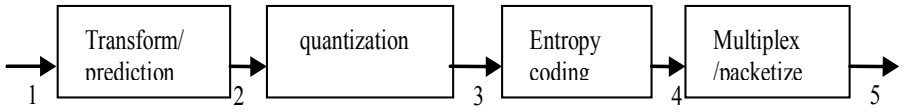


FIGURE 2: Candidate Domains Used to Apply Encryption to Multimedia.

According to Fig. 2, there are two straight forward places to apply generic encryption to multimedia. The first possibility is to encrypt multimedia samples before any compression, stages 1 and 2, Qiao et al. [4] and Uehara and Safavi-Naini [5] are examples of pre-compression selective encryption. The main problem with this approach is that the encryption often significantly changes the statistical characteristics of the original multimedia source, resulting in much reduced compressibility. Cheng and Li, 2000. The wavelet-based compression algorithm SPIHT [6] is an example of post-compression encryption scheme, stage 4 and 5. Wu et al proposed encryption scheme based on encoding with multiple Huffman tables (MHT) used alternately in a secret order [1]; is an example of in-compression selective encryption stages 3, and 4. The encryption with reasonably high level of security and unaffected compression can be achieved simultaneously, requiring almost negligible additional overhead. One of the major advantages by using this kind of joint encryption-compression approach is that encryption and compression can be achieved in one single step, which simplifies the system design an makes it flexible for some advanced multimedia processing such as scalability and rate shaping.

2.1. Multiple Huffman Tables (MHT) Technique

The MHT algorithms [1][7]-[9], aiming to increase the model space while maintaining the computational efficiency, keep the structure of the Huffman tree but enlarge the model space through tree mutation. The procedure of the basic MHT algorithm is described as follows:

Step1: Train four original Huffman trees from different sets of training data. e.g. Huffman table of the JPEG DC coefficients.

Step2: Based on the original trees, perform tree mutation, to create the whole Huffman tree space.

Step3: Randomly select m different tables from the space, and number them from 0 to m-1.

Step4: Generate a random vector P = $\{P_0, P_1, \cdots, P_{m-1}\}$ each p is an Integer ranging from 0 to m-1.

Step5: For the i^{th} encountered symbol, use table $P_{i\ (modn)}$ to encode it.

MHT coding [1] makes use of standard coding tables. It is included in the final bit-stream for every image to be compressed. This approach presents disadvantages:

1. Visual degradation: very high-visual degradation can be achieved.

2. Cryptographic security: Gillman and Rivest [10] showed that decoding a Huffman coded bitstream without any knowledge about the Huffman coding tables would be very difficult. However, the basic MHT is vulnerable to known and chosen plaintext attacks as pointed out in [11].

3. It writes all codes of the corresponding tables in the final bitstream even if only some of they were used to encode the associated events of the particular input image.

4. It does not make use of any statistic about the distribution of the events of the image.
   To improve the security several kinds of enhanced MHT schemes have been proposed:
   - By inserting random bit in the encrypted bit stream or integrating with a stream cipher [8].
   - Recently another scheme via random rotation in partitioned bit streams has been reported [9].

3. OPTIMIZED MULTIPLE HUFFMAN TABLES (OMHT)

OMHT compression-encryption technique is a modification to the MHT scheme; it generates different Huffman tables for each type of images instead of using fixed Huffman tables for all images as in MHT technique. The main advantage of using OMHT technique over other lossy compression technique is that it produces a much smaller compressed file than any compression method, while still meeting the advantage of encryption. Remove small, invisible parts, of the picture is based on an accurate understanding of how the human brain and eyes work together to form a complex visual system. As a result of these subtle reductions, a significant reduction in the resultant file size for the image sequences is achievable with little or no adverse effect in their visual quality. As shown in Fig. 3 OMHT process takes two parallel paths A, and B, so it takes no additional time to add encryption to the compressed bitstream as both traditional and selective encryption techniques.

3.1. The Procedure of Compressing the Original Image

As shown in Fig. 3 (path A), The input *N*x*M* image is first converted into single vector by concatenating successive rows beside each other to form a long row that contains all the image pixels using matrix to vector converter. This vector is exposed to DCT to transform the image from spatial domain into frequency domain in which energy of the image information is concentrated in a few number of coefficients. The output of the DCT process is a vector that has the same length of the image (number of pixels in the image), but with many values approximated to zeros. After applying the DCT the output coefficients are arranged in a descending order according to its energy content. The energy content of the coefficients is summed from the beginning of the vector and toward the end till a specific energy percentage (*EP*) of the image energy is reached. Those coefficients that carry *EP* energy percent are chosen to be transmitted and the rest coefficients are neglected since they carry only very small energy that will not affect the visual quality of the recovered image. This *EP* value depends on image characteristics and it can be varied to achieve the desired compression ratio and the signal to noise ratio according the application: As we decrease the *EP* value, a higher compression ratio is obtained with slightly lower signal to noise ratio. Now the number of the transmitted coefficients (*Tc)* becomes very small. The reduced coefficients vector returned back to the spatial domain using IDCT to be processed by an efficient quantizer.

The proposed Least Probable Coefficients Approximation (LPCA) quantizer as shown in Fig. 4(a) and its flowchart in Fig. 4(b) reduces the output values of the IDCT by calculating their occurrence probabilities. The IDCT coefficients are arranged in a descending order according to their probabilities in a vector. The desirable quantization levels are taken as the most probable coefficients from the beginning of the arranged vector; if the required CR and SNR are achieved by transmitting only four quantization levels, those quantization levels are the first four coefficients in the arranged vector. The probability of the last *QL* is called neglecting probability (*NP*).

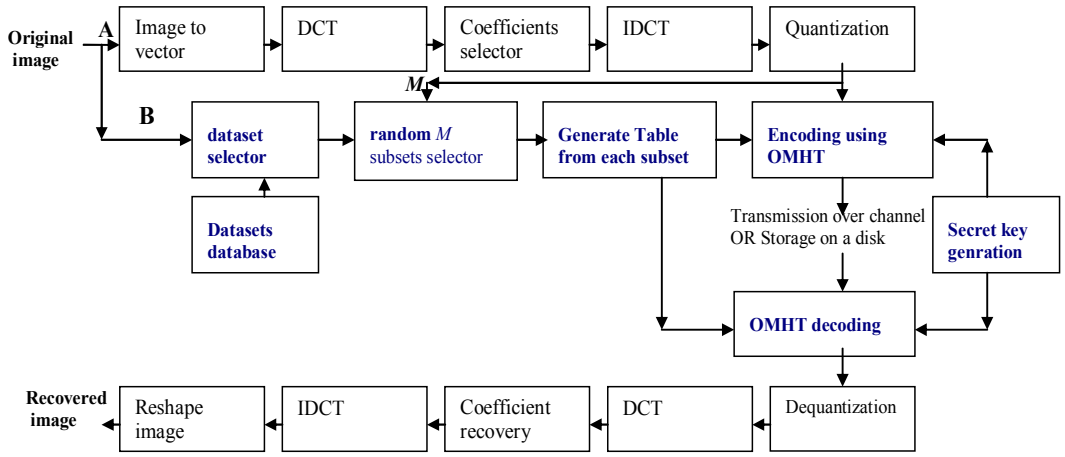


FIGURE 3: Optimized Multiple Huffman Table (OMHT) Coding System

All coefficients with probability less than *Np* are approximated to the nearest quantization level in value. This technique is irreversible; this means that the dequantized values can't be turned back to their original values leading to quantization losses. The quantization procedures are summarized as follows:

- IDCT coefficients are arranged in a descending order according to their probabilities in a vector.
- The desirable *n* quantization levels are taken as the *n* most probable coefficients from the beginning of the arranged vector.
- The probability of the last *QL* is called neglecting probability (*NP*).
- All coefficients with probability less than *Np* are approximated to the nearest quantization level in value. The proposed quantization reduces number of transmitted values but not the number of transmitted coefficients.

After the transmitted values are reduced by quantization, each quantized level is assigned a codeword using Huffman encoder that enables representing an image in a more efficient way with smallest memory for storage or transmission. Huffman coding is used to code the quantized values statistically according to their probability of occurrences. Short code words are assigned to highly probable values and long code words to less probable values. The average number L_{avg} of bits required to represent a symbol is defined as,

$$L_{avg} = \sum_{k=1}^{L} I(r_k) P(r_k)$$

Where, rk is the discrete random variable for k=1,2,…L with associated probabilities P(rk). The number of bits used to represent each value of rk is I(rk). The number of bits required to represent an image is calculated by number of symbols multiplied by L_{avg} [9].

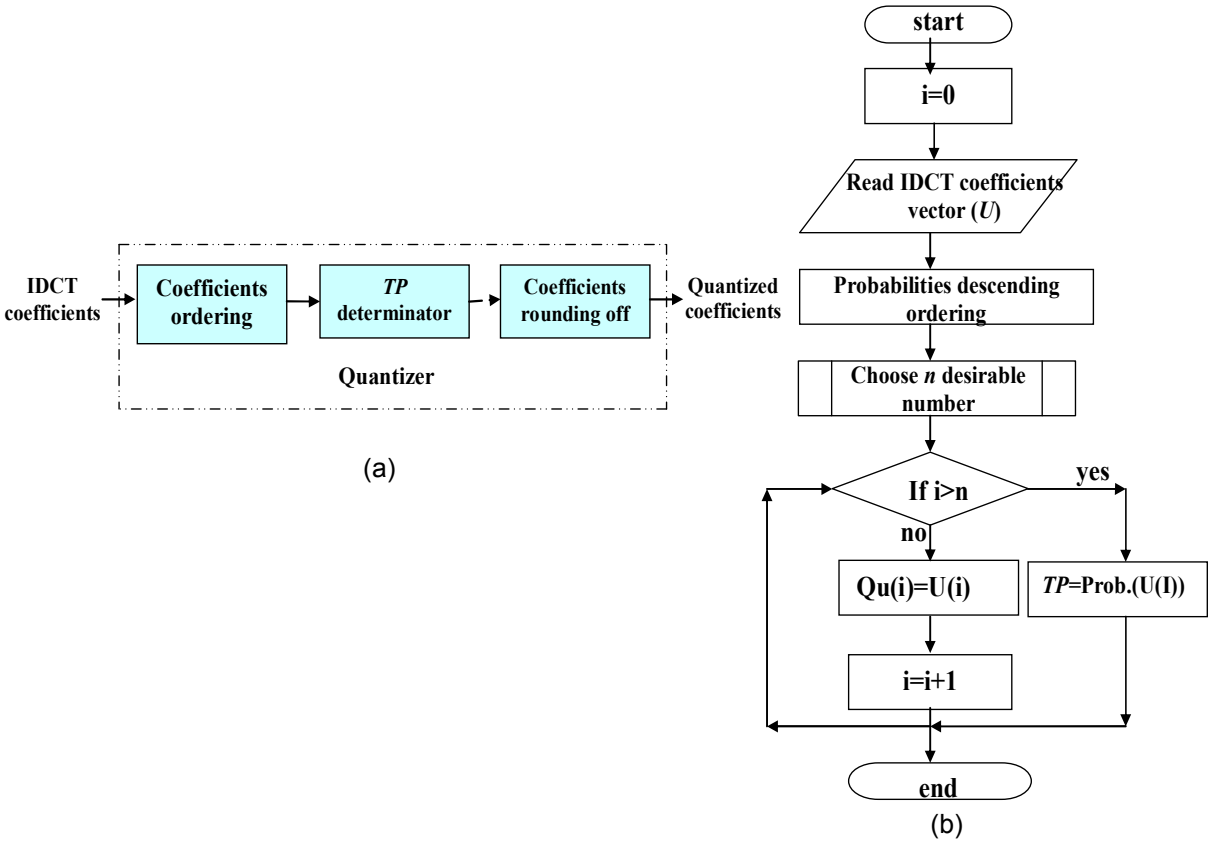


FIGURE 4: (a) Block Diagram of LPCA Quantizer (b) LPCA flowchart

3.2. Procedure of Preparing and Using the OMHT Tables

Following is the procedure of preparing and using the optimized multiple Huffman tables and how it is used to both encode and encrypt images as shown in Fig. 3 (path B).

Step 1: images training set are divided into L datasets. Each dataset's images have the same properties.

Step 2: each dataset contains N images.

Step3: The input image compared to datasets to select the dataset that has the same properties.

Step4: randomly choose M subsets each subset contains K images from the dataset. Concatenate all images of each subset and calculates the pixels probabilities. Then draw Huffman tree and find the Huffman table contains the different pixels' values and their associated variable codewords. Now we have M different tables to be used.

Step5: Tables are saved at each decoder, and the order by which the tables are generated and used is kept secret.

Step6: Number the generated M tables from 0 to M-1.

Step7: Generate a random vector P (the secret order) its length equal to the length of image under consideration. Each element value in P ranges from 0 to M-1.

Step8: For the ith encountered symbol (coefficient to be encoded), use table P i(mod n) to encode it.

4. PERFORMANCE ANALYSIS of OMHT

For performance evaluation, the following experiments measure the compression performance and encryption strength of OMHT using test images that contains gray and colored images. The compression performance of OMHT is analyzed by calculating the compression ratio (CR), number of bits per symbol (BPP), the peak signal to noise ratio (PSNR), and the mean square error (MSE). A comparison between the proposed scheme based on generating tables based on statistical modeling of large dataset for each types of images, and a compression using fixed predetermined encoding tables, JPEG standard, on which the MHT technique based on done to show the effectiveness of the proposed scheme in compression. The encryption strength of the OMHT is tested and compared with other encryption techniques.

The achieved compression ratio can be calculated from the following equation:

$$CR = \frac{original}{compressed}$$

Where the original is the size of the original image and the compressed is the size of the Huffman encoder output compressed bitstream. Calculate the bit per pixel (BPP) is defined as:

$$BPP = \frac{B}{P}$$

Where *P* is the total number of pixels in an image and *B* is the total number of transmitted bits for this image. As a measure of reconstructed image quality, the peak signal-to-noise ratio (PSNR) in dB is used, this is defined as follows:

$$PSNR_{dB} = 20 \log_{10} \frac{2^n - 1}{\sqrt{MSE}}$$

Both mean square error (MSE) and the signal to noise ratio (SNR) for an *n*X*n* image are calculated from the following equations:

$$MSE = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha[i,j] - \beta[i,j])^2$$

$$SNR = 10 \log_{10} \times \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha[i,j]^2)}{\sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha[i,j] - \beta[i,j])^2}$$

Where, α[*i,j*] and β[*i,j*] denote the original and decoded levels of the pixel [*i,j*] in the image, respectively. A larger PSNR value means that the encoded image preserves the original image quality better.

Experiment 1 uses lossy OMHT to encrypt and compress the Lena image. It gives the ability to control the compression ratio and peak signal to noise ratio by either change number of *QL* while the amount of *Tc* is constant, or changing the amount of *Tc* while number of *QL* is constant.

As shown in Table 1, and Fig. 5, and 6, while the number of quantization levels is constant at *q*=128 and the amount of transmitted DCT's coefficients changes from *Tc*=98.5% of the image energy to *Tc*= 99.9%. As *Tc* increases, the CR decreases providing an increase in PSNR.

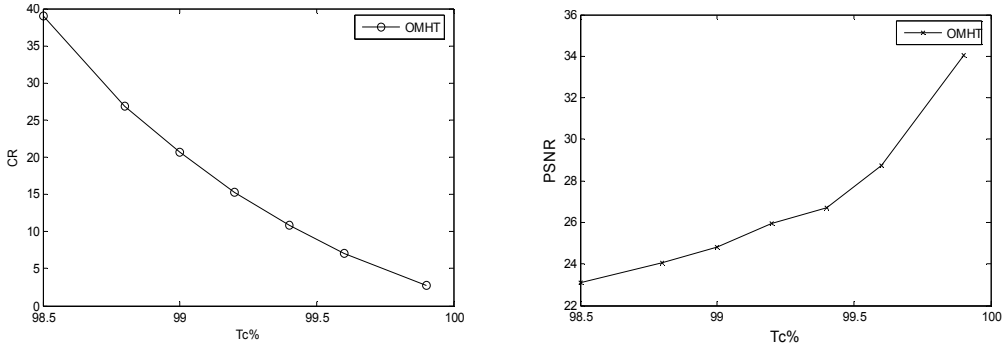
(a) Original Image (b) Decoded for *TC*=99.9% (c) Decoded for *TC*= 99.6% (d) Decoded for *TC*=99.4%

(e) Decoded for*TC*=99.2% (f) Decoded for *TC*= 99% (g) Decoded for *TC*= 98.8% (h) Decoded for *TC*=98.5%

FIGURE 5: The Effect of Reducing Number of Transmitted Coefficients on Image Visual Degradation

	Tc=99.9%	***Tc=99.6%***	***Tc=99.4%***	***Tc=99.2%***	***Tc=99%***	***Tc=98.8%***	***Tc=98.5%***
CR	2.8179	7.0518	10.860	15.252	20.612	26.936	38.973
BPP	2.8390	1.1345	0.7366	0.5245	0.3881	0.2970	0.2074
PSNR	34.032	28.744	26.7053	25.929	24.794	24.065	23.125
MSE	20.848	94.222	116.673	179.287	200.80	290.092	360.09
SNR	26.947	21.669	19.6207	18.8446	17.719	16.9809	16.040

TABLE 1: Compression Performance of Applying OMHT with Different Number of Transmitted Coefficients



(a) Variation of CR with Varying *Tc* (b) Variation of PSNR with Varying *Tc*

FIGURE 6: The Effect of Reducing OMHT Number of *Tc* on CR, and PSNR

OMHT provides the ability to maintain *Tc* constant and varies the number of *QL*. As shown in Table 2, and Fig. 7, and 8, while the amount of transmitted coefficients is constant at *Tc*=99.5% and the number of quantization levels changes from using four quantization levels to using 256 quantization levels. As the number of quantization levels increases, the compression ratio decreases providing an increase in peak signal to noise ratio.

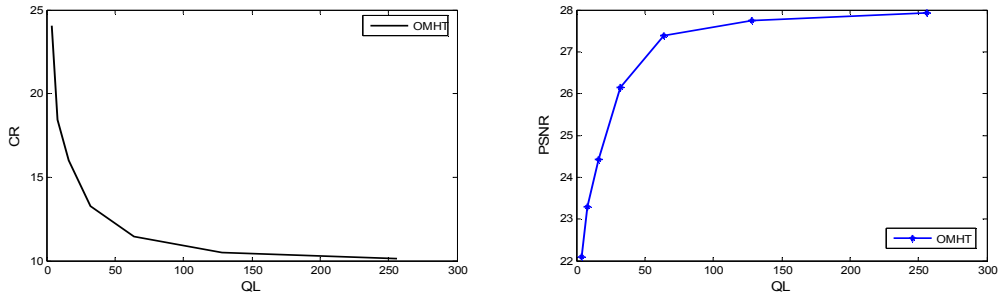
(a) Original Image (b) Decoded for q=256 (c) Decoded for q=128 (d) Decoded for q=64

(e) Decoded for q=32 (f) Decoded for q=16 (g) Decoded for q=8 (h) Decoded for q=4

FIGURE 7: The Effect of Reducing Number of Quantization Levels on Image Visual Degradation

	Q=4	***Q=8***	***Q=16***	***Q=32***	***Q=64***	***Q=128***	***Q=256***
CR	24.08	18.437	15.99	13.25	11.45	10.49	10.14
BPP	0.332	0.4339	0.500	0.604	0.699	0.762	0.789
PSNR	22.09	23.295	24.42	26.14	27.59	27.75	27.92
MSE	382.2	305.51	222.2	119.67	102	98.12	80.94
SNR	15	16.21	17.33	19.06	20.5	20.66	20.84

TABLE 2: Compression Performance of Applying OMHT with Different *QL* on Lena Image



(a) Variation of CR with Varying *QL* (b) Variation of PSNR with Varying *QL*

FIGURE 8: The Effect of Reducing OMHT Number of Quantization Levels on CR, And PSNR

Experiment 2 compares the compression performance of lossy OMHT with that of lossy JPEG technique to prove that the proposed technique adds security without affecting the compression ratio or the PSNR. Table 3 provides a comparison of CR between OMHT, and JPEG at Different BPP on Lena Image, while Table 4 provides a comparison of PSNR between OMHT, and JPEG at Different BPP on Lena Image. From Tables 3 and 4 it is obvious that using lossy OMHT technique provides higher PSNR and storage space and transmission bandwidth required than JPEG especially at low bitrates.

BPP	OMHT	JPEG
0.2	39	39.01
0.18	44.4	43
0.16	49.4	49
0.14	57.4	58.02
0.12	65.6	65.3
0.1	73.9	73

TABLE 3: Comparison of CR between OMHT, and JPEG at Different BPP on Lena Image

BPP	OMHT	JPEG
0.2	22.6	21.14
0.18	22.2	20
0.16	21.9	19.4
0.14	21.6	18
0.12	21.3	16.7
0.1	21	15

TABLE 4: Comparison of PSNR between OMHT, and JPEG at Different BPP on Lena Image

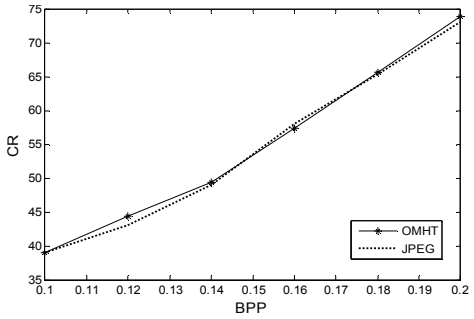


FIGURE 9: CR of both OMHT technique and JPEG for Lena image

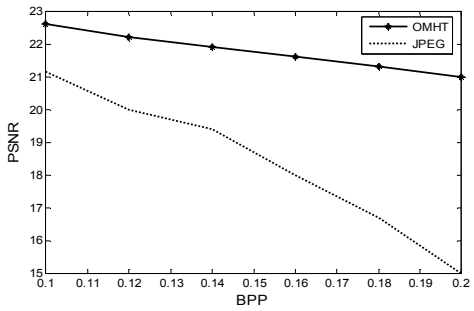


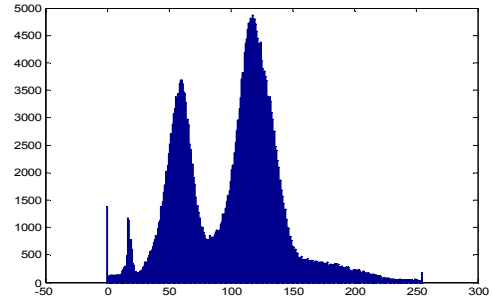
FIGURE 10: PSNR of both OMHT technique and JPEG for Lena image

Fig. 9 shows that both OMHT technique and JPEG technique have nearly the same compression levels at the same number of bits per pixel for Lena image. Fig. 10 shows that PSNR of OMHT technique is higher and more stable at low bitrate than that of JPEG for Lena image.

Experiment 3 measures the encryption strength performance of the proposed OMHT technique, colored football image in RGB (288x352x3) shown in Fig. 11(a) with its histogram shown in Fig. 11(b) is compressed and encrypted using the OMHT uses multiple Huffman tables, generated from a large set of training images that have the same type of the test image used in a secret order (secret key). Fig. 11(c) and 11(d) shows the test image and its histogram after decoding it with another technique as JPEG. While Fig. 11(e) and 11(f) shows the test image and its

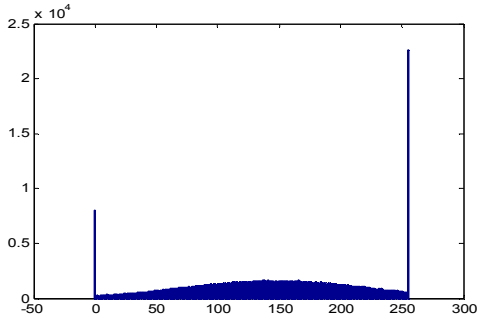
histogram after decoding it with OMHT technique and the same encoding tables but without knowing the secret order (secret key). It is obvious that OMHT provides high perceptual security

(a) Original Image

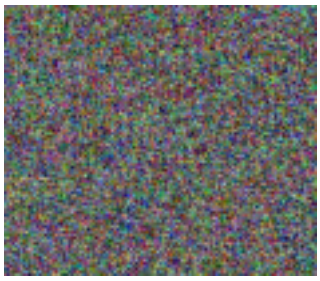


(b) Histogram of Original Image

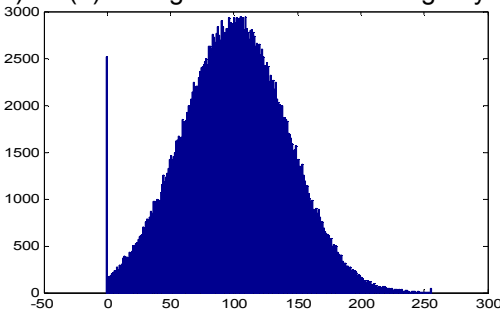
(c) Decoded image with JPEG(PSNR=5dB)



(d) Histogram of decoded Image by JPEG



(e) Decoded with OMHT wrong key (PSNR=5.7dB)



(f) Histogram of decoded Image by OMHT

FIGURE 11: The Effect of Decoding Football Image without the Secret Order

Fig.12 shows the perceptual performance comparison between OMHT and other different encryption techniques used to encrypt Lena image. Fig.12(a) shows the original Lena image, Fig.12(b) shows the decoded image that was encrypted by OMHT, Fig.12(c) shows the decoded image that was encrypted by building a three level pyramid and encrypting the lowest resolution plus the first residual (HP Mode 30% encryption), Fig.12 (d) shows the decoded image that was encrypted by encrypting only the DC coefficients with the first AC coefficient of each block (SS Mode 30%), Fig.12 (e) shows the decoded image that was encrypted by scrambling the DC coefficients and one bitplane or three bitplanes (MM Mode 30%), Fig.12(f) shows the decoded image that was encrypted by encrypting the most significant bits of all coefficients (SA Mode 30%), Figs.2.band3.b clearly show that there can be still information left in the unencrypted parts of the data after selective encryption has been applied, Fig.12 (g) shows the decoded image that was Encrypted by Run-length, Fig.12 (h) Encrypted by sign bit encryption, Fig.12 (i) shows the decoded image that was Encrypted by band permutation(10 bands), Fig.12(j) shows the decoded image that was encrypted by bitplane permutation(n=6), Fig.12(k) shows the decoded image that was encrypted by bitplane permutation (n=7), and Fig.12(l) shows the decoded image that was encrypted by MHT. It is obvious that the PSNR of the decoded image that is encrypted by OMHT

is smaller than it is in all other techniques. So, the perceptual security strength of the OMHT technique is higher than other techniques.

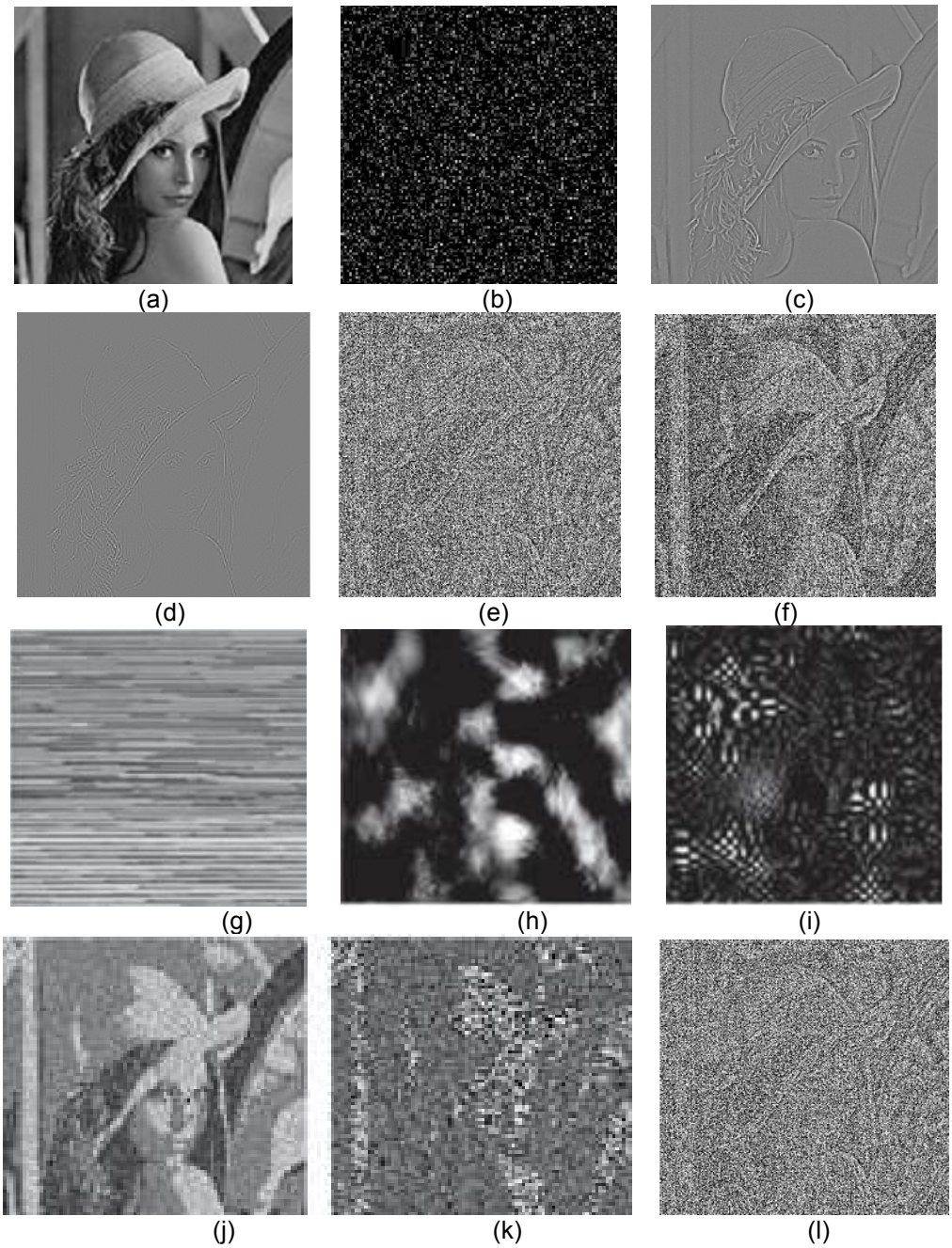


FIGURE 12: The Effect of Decoding Lena Image Encrypted with Different Techniques by JPEG: (a) Original Image, (b) Encrypted by OMHT(PSNR=4.8 dB), (c) Encrypted by HP Mode (PSNR=14.7 dB), (d) Encrypted by SS Mode(PSNR=14.2 dB), (e) Encrypted by MM Mode (PSNR=6.2 dB), (f) Encrypted by SA Mode(PSNR=6.4 dB), (g) Encrypted by Run-length (PSNR=6.5 dB), (h) Encrypted by sign bit encryption (PSNR=6.1 dB), (i) Encrypted by band permutation(10 bands) (PSNR=7.23 dB) (j) Encrypted by bitplane permutation (n=6) (PSNR=13.8 dB), (k) Encrypted by bitplane permutation (n=7) (PSNR=9.18 dB), (l) Encrypted by MHT (PSNR=6.4 dB),

5. COMPUTATIONAL COST ANALYSIS

The evaluation of the computational speed of ciphers usually consists of the analysis of the key-setup cost, the encryption cost and the decryption cost [16]. The encryption and the decryption costs are usually similar, and they are more important than the key-setup cost because one single key-setup can often be followed by thousands of encryption/decryption operations. In the following, we analyze these costs of our OMHT encryption scheme, and compare them with those of MHT and modern ciphers.

a) Key-Setup cost: The key-setup process includes all the computation and memory allocation operations prior to actual encryption of the first bit in the plaintext. The computational cost of OMHT key-setup is dominated by the construction of optimized multiple Huffman tables, generation of the secret order by which those tables are used, and comparing the test image with datasets. OMHT takes about 10 operation per table generation, single operation for secret key generation, and *L* operation for comparison. The total number of operations equal 10X*M*X*L*+1+*L,* where *L*, M is number of datasets and number of subsets respectively. For *L*=4, *M*=20, the net Key-Setup cost =805 operations. For MHT technique it takes 20 operations per table entry, the total cost would be 20x*t*x*m*, where t and m are the table size and the number of selected tables, respectively. For the example of JPEG dc coefficient encryption as shown in the previous subsection, the key-setup cost would be around 2000 operations (*t*=13 and *m*=8).Compared with the ciphers listed in Table 6,the key-setup cost of OMHT encryption is much smaller than MHT and other ciphers.

b) Encryption/Decryption cost: The net computational cost of the OMHT is the same as the basic MHT-encryption scheme [1] is less than one CPU operation per encrypted bit as explained below. When a symbol is to be encoded with a normal Huffman coder, the shift amount is added to the base address of the table to obtain the address of the desired Huffman code. This process is illustrated in Fig.13 (a). In the basic MHT system, we store the base addresses of the tables in a cyclic queue according to the order that they are used. When a symbol is to be encoded/encrypted, the base address is first loaded from the memory, and then the shift-amount is added to it. Afterwards, the index to the cyclic queue of base addresses should be increased by one. Then, the index should be compared with the end of the queue in order to decide whether it should be reset to the beginning of the queue. Therefore, the computational difference between our cipher/encoder and a normal Huffman coder is one memory-load, one addition and one comparison operation for each symbol encoded. The encoding process of the proposed cipher/encoder is shown in Fig.13 (b). Since each symbol in the original data usually corresponds to more than 3 bits in the Huffman bitstream, then encryption cost of our algorithm is less than one CPU operation per encrypted bit, which is around 20 times smaller than the well-known AES as listed in Table 6.

Recently, a new cryptographic cipher named COS [18] with a very fast speed is gaining popularity. It is around 4–5 times faster than AES. Compared to COS, the encryption cost of OMHT is still several times smaller.

Cipher Type	Key-setup Cost (CPU instructions)	Encryption Cost (CPU instructions/bit)
MARS	9416	25
RC6	10372	22
Rijndael	35484	20
Serpent	26308	28
Twofish	37692	20

TABLE 6: Computational Costs of AES Finalists on a Pentium-MMX Machine. The Figures in This Table are Translated from [17] by Assuming Two CPU Instructions are Executed in Every Clock Cycle in a Pentium-MMX CPU

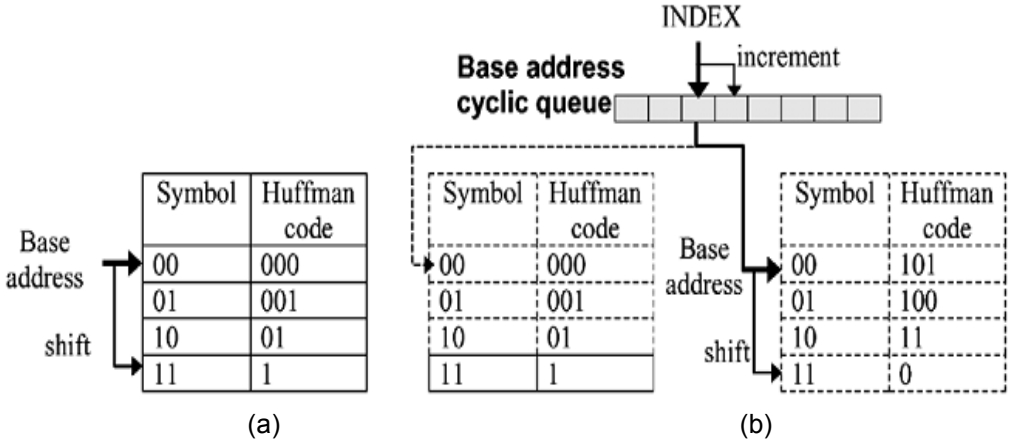


FIGURE 13: (a) Normal Huffman Coder Adds the Shift Amount to the Base address of the Table to Obtain the Address of the Desired Huffman Code. (b) OMHT Loads the Base Addresses of Huffman Tables from a Cyclic Queue, and the Index to the Queue is Increased by One After Coding of Each Symbol.

6. CONCLUSIONS

The experiments' results reveal that the proposed OMHT technique achieves better compression and security performance than that of MHT, and JPEG Image Compression Standard especially at low bitrate. The OMHT scheme provides

- **High security:** resistance against various types of attacks, including the ciphertext-only attack and the known/chosen plaintext attack[19].
- **Low encryption cost:** the encryption cost not exceed very small portion of the total computation cost of compression
- **No harm to the compression ratio:** The increase of the final bit stream size due to encryption is not higher than 0.5% of the original coded bitstream.
- Joint compression-encryption OMHT technique achieves both high security and compression performance in one single step, which simplifies the system design and reduces time required to perform compression followed by encryption.
- Since images have different statistics, using the same fixed JPEG standard predefined coding tables as suggested in MHT technique will not be effective in encoding all image and video types.

- The OMHT method obtains better performance in terms of storage space use and more stable peak signal to noise ratio than that of JPEG in encoding an image with small and great gray-level variations among adjacent pixels.
- Receivers haven't the secret order cannot decode the encoded images successfully.
- Further, the proposed new compression-encryption technique could be applied on any source data, not only images, which uses Huffman coding to achieve better compression ratio. Therefore, the proposed technique will be suitable for compression of text, image, and video files.

7. REFERANCES

1. C.-P. Wu and C.-C. J. K. Kuo. "*Design of integrated multimedia compression and encryption systems*". IEEE Transactions in Multimedia, vol. 7, no. 5, pp. 828–839, 2005.

2. W. Stallings. "*Cryptography and Network Security Principles and Practices*", Upper Saddle River, NJ: Prentice Hall, 2003.

3. M. Van Droogenbroeck and R. Benedett. "*Techniques for a selective encryption of uncompressed and compressed images*". In Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS '02), pp. 90–97, Ghent, Belgium, September 2002.

4. L. Qiao, K. Nahrstedt, and M.-C. Tam."*Is MPEG encryption by using random list instead of zigzag order secure?*". in Proceedings of the IEEE International Symposium on Consumer Electronics (ISCE '97), pp. 226–229, Singapore, December 1997.

5. T. Uehara and R. Safavi-Naini."*Chosen DCT coefficients attack on MPEG encryption scheme*". in Proceedings of IEEE Pacific Rim Conference on Multimedia, pp. 316–319, Sydney, Australia, December 2000.

6. H. Cheng and X. Li. "*Partial encryption of compressed images and videos*". IEEE Transactions on Signal Processing, vol. 48, no. 8, pp. 2439–2451, 2000.

7. C.-P. Wu and C.-C. Kuo. "*Efficient multimedia encryption via entropy codec design*". Proc. SPIE, vol. 4314, Jan. 2001.

8. D. Xie and C. J. Kuo. "*Enhanced Multiple Huffman Table (MHT) Encryption Scheme Using Key Hoping*". In Proceedings of IEEE International Symposium on Circuits and Systems, pp.568–571, May2004.

9. D. Xie and C. J. Kuo. "*Multimedia Data Encryption via Random Rotation in Partitioned Bit Stream*". In Proceedings of IEEE International Symposium on Circuits and Systems, pp.568–571, May2004.

10. D. W. Gillman and R. L. Rivest. "*On breaking a Huffman code*". IEEE Transactions on Information Theory, vol. 42, no. 3, pp. 972–976, 1996.

11. J. Zhou, Z. Liang, Y. Chen, and O. C. Au. "*Security analysis of multimedia encryption schemes based on multiple Huffman table*". IEEE Signal Processing Letters, vol. 14, no. 3, pp. 201–204, 2007.

12. W. Pennebaker and J. Mitchell. "*JPEG Still Image Data Compression Standard*", Van Nostrand Reinhold, New York, 1993.

13. http://www.jpeg.org (JPEG resources) [accessed at 4/8/2010]

14. http://www.jpeg.org/public/jfif.pdf (JPEG file interchange format) [accessed at 8/8/2010]

15. (independent JPEG group) ftp.uu.net:/graphics/jpeg [accessed at 8/8/2010]

16. C.-P. Wu and C.-C.J. Kuo. "*Efficient multimedia encryption via entropy codec design*". In Proc. SPIE Int. Symp. Electronic Imaging 2001, vol. 4314, Jan. 2001, p.128.

17. J. Nechvatal et al. "*Report on the Development of the Advanced Encryption Standard*". National Institute of Standards and Technology, U.S. Dept. Commerce, Tech. Rep., Oct. 2000.

18. E. Filiol and C. Fontain. "*A new ultra fast stream cipher design: COS ciphers*". In Proc. 8th IMA Conf. Cryptography and Coding, Dec. 2001.

19. Shaimaa A. El-said, Khalid F. A. Hussein, and Mohamed M. Fouad. "*Securing Multimedia Transmission Using Multiple Huffman Tables Technique*". Electrical and Computer Systems Engineering Conference (ECSE'10), Egypt, 2010.