International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 2, February- 2013

A Survey of Video Encryption Algorithms Implemented In Various Stages of Compression

S. Rajagopal
Asst. Professor
National Engineering College

Dr. A. Shenbagavalli
Professor
National Engineering College

Abstract

Video data security is very important for multimedia commerce on the internet and real-time video multicast. However, traditional encryption algorithm for data secrecy such as DES, AES may not be suitable for multimedia applications because they are unable to meet the real-time constraints required by the multimedia applications. For video applications lightweight encryption algorithms are suitable. This paper analysis the possibility of deploying encryption algorithm in various stages of compression. This joint compression and encryption of video data provides security for real time applications like video conferencing, surveillance camera data protection, etc. From the generic structure of a video encoder, the analysis of incorporating several encryption algorithm in the compression stages like transformation stage, in coding stages were presented.

1. Introduction

In this survey we present an overview, classification and evaluation of video encryption algorithms, a topic to which numerous proposals have been made. The survey focuses duly on both common video compression format and also on video encryption algorithms that can be implemented during the process of video compression itself. The use of video applications such as the World Wide Web and video conferencing has increased dramatically in recent years. The encryption of video data is also performed to protect the privacy of the users. Traditionally, an appropriate compression algorithm is applied to the multimedia data and its output is encrypted by an independent encryption algorithm. This process must then be reversed by the decoder. The processing time for encryption and decryption is a major bottleneck in real-time video communication and processing.

Traditional cryptographic algorithms for data security are often not fast enough to process the vast amount of continuous data, such as video, to meet the real-time constraints. Most algorithms either added huge amount of overheads to both the encoding and decoding process. The various encryption algorithms that can be implemented at specific stages of video compression are analyzed. The various video encryption algorithms which encrypts Discrete Cosine Transform (DCT) co-efficients, Difference Motion Vectors (DMV) and at coding (VLC, EC) stages of compression are analyzed in the rest of the paper.

2. Analysis of Encryption at Compression Stage

Video compression and encryption are associated process in secure multimedia systems and applications. Some video encryption algorithms are even fully embodied in a video codec. Standardized video compression technologies like MPEG-1(ISO/IEC, 1993), MPEG-2(ISO/IEC, 2000), H.261(ITU-T, 1993), H.263(ITU-T Recommendation H.263, 1998) and MPEG-4/H.264 AVC(Advanced Video Coding) (ITU-T Recommendation H.264, 2007; ISO/IEC, 2005) are widely deployed for economically storing digital videos on storage constrained devices or efficiently transmitting them over bandwidth-limited networks. Video compression can be viewed as compression of

sequence of images or in other words, image compression with temporal component. All video coding standards utilize the hybrid coding approach, i.e. they compress video data by using intra-frame and inter-frame coding simultaneously. A hybrid video encoder firstly removes temporal redundancy, then removes spatial redundancy and finally removes statistical or entropic redundancy. When transmitting set of pictures it is possible for data to be lost or computed, leading to accumulative errors in prediction. The intra-frame coding is used to reduce spatial redundancy that exists within the frame. It compresses an entire video frame independently of any other frames. The resulting coded frame is denoted as I-frame. Temporal redundant is not removed in Intra (I-picture). Every group of pictures (commonly every 12) another I-picture is formed, while intermediate pictures are predicted (P-pictures) or bi-predicted (B-pictures) from previous and succeeding pictures. Although there are differences in the concrete coding algorithms applied, the compression standards are built upon the same fundamental set of function elements. A generic structure of a video encoder is shown in Fig. 1.

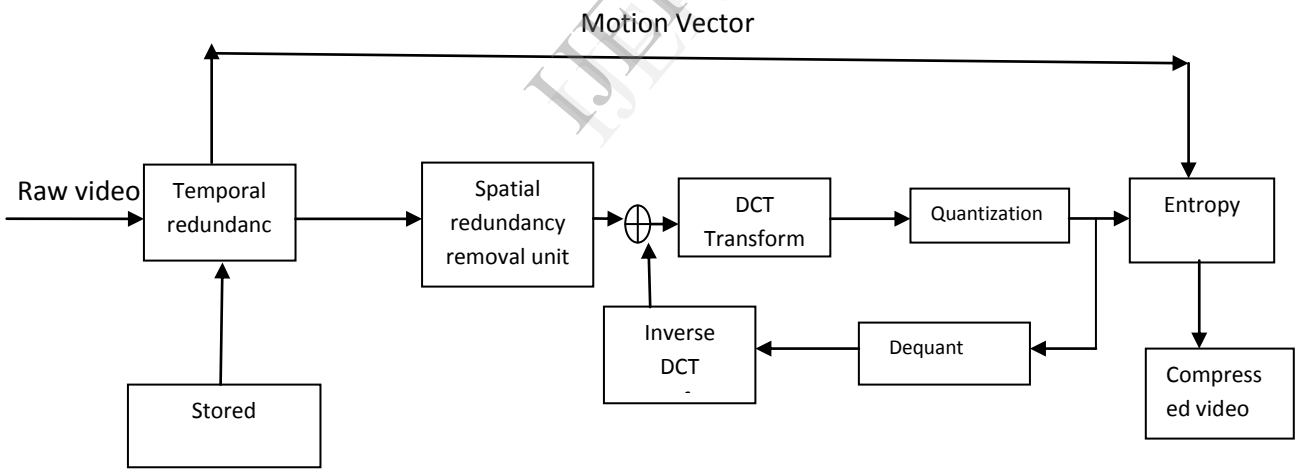

Fig.1: Generic structure of a video encoder

3. Transform Co-efficient , MV Encryption

3.1 DCT coefficients scrambling encryption algorithm

Yan li & Main Cai (2009) proposed encryption of videos by scrambling discrete cosine transform (DCT) coefficients in a 16*16 macroblock. Main Cai encryption algorithms are: Complete scrambling algorithm, Subsection scrambling algorithm, High-low-frequency scrambling algorithm, Sub blocks scrambling algorithm. Complete scrambling algorithm disorders DCT coefficients in a macroblock· Non-zero coefficients are about 1/3 of total coefficients after discrete cosine transform. Subsection scrambling algorithm divides DCT coefficients and scrambles them in different segments according to security and compression ratio. The encryption DCT coefficients scrambling algorithm only breaks the order of coefficients or macroblocks and doesn't encrypt video data, it has low security. High-low-frequency scrambling algorithm's security is general, subsection scrambling algorithm's is lower than complete scrambling algorithm's and sub blocks scrambling algorithm's is the lowest. Scrambling time is shorter than encryption time, so DCT coefficients scrambling algorithm has high speed.

2 DCT coefficients encryption algorithm and MVD signs encryption algorithm

DCT coefficients and MVD (the motion vectors differences) are the important data in H.264 compressing and encoding process. DC coefficients contain main information of a video, AC coefficients contain detail information, and

MVD contain dynamic information. There are DCT coefficients selective encryption algorithm, DCT coefficients signs encryption algorithm and MVD signs encryption algorithm.

DCT coefficients selective encryption algorithm chooses appropriate DC, AC coefficients to encrypt, its security is high, but selecting data and encrypting operation reduced its speed. DCT coefficients signs encryption algorithm extracts DCT coefficients signs to be binary bit streams and XOR them with secret keys, its security is not high, but only encrypt signs increase its speed. MVD signs encryption algorithm extracts MVD signs to be binary bit streams and XOR them with secret keys, it have low security but high speed too.

Encrypting DCT coefficients has destroyed data's statistical characteristics and changes the compression ratio. Only encrypting the signs algorithms have small affection to compression ratio. They all don't encrypt the header information and all the DCT coefficients encryption algorithms have date operability.

3.3 The Selective Video Encryption Algorithm

Saranya and Varalakshmi (2011) proposed selective video encryption. The main aim is to select proper data before compression for encryption which gives higher efficiency and lower cost. In the H.264 standard, a macroblock is predicted either spatially or temporally.

A macroblock is composed of luminance and chrominance samples. In their encryption method, luminance transform coefficients (LTC) are considered to encrypt in order to minimize the computational burden. In proposed LTCE algorithm, the one DC LTC must be encrypted. Moreover, the DC is independent of the other AC coefficients. The DC encryption and AC encryption are independent of each other, and they can be freely combined. This paper proposes a single scheme with RC4 stream ciphers. RC4 generates a pseudorandom stream of bits in which the encryption is combined with the plaintext using bit-wise exclusive-or; decryption is performed the same way.

3.4 Motion vector encryption algorithm

Zheng Liu and Xue Li (2004) proposed Motion Vector Encryption Algorithm. In this paper, a novel algorithm for encrypting video motion vector is presented. They considered the static features in motion vectors and hide the similarities among motion vectors. Firstly the motion vectors are XORed with a random iterator to make individual motion vectors not detectable. The process is called concealing. Secondly they scramble motion vectors with random generated numbers to distance them in the spatial domain. Thus motion vectors will not be cluster able by attackers. This phase is called distancing. Both processes are of the same importance for encrypting motion vectors. The security of the system depends on the randomness of the random number. By choosing correct random number generation algorithm, there will be no statistical cryptanalysis method available for attackers. As such the system security can be guaranteed.

4. Secured Advanced Video Coding

Shiguo Lian and Zhong Xuan Liu (2006) proposed Secured Advanced Video Coding (AVC) encoding scheme. During AVC encoding, sensitive data as intra-prediction mode residue data and motion vectors are partially encrypted. In their proposed method, the intra-prediction mode is encrypted based on Exp-Golomb entropy coding, the intra-macroblocks DCs are encrypted based on context based adaptive variable length coding (CAVLC) and intra-macroblocks ACs and the inter-macroblocks. MVDs are sign-encrypted with a stream cipher followed with variable length coding. The intra-prediction stage output is first

4.1 The Exp-Golomb Encryption Algorithm (EGEA)

This encryption algorithm encrypts the intra- prediction modes during variable length coding, realizes encryption and variable-length coding at the same time, and keeps the codeword's length unchanged. The intra-prediction stage output is first encoded into a variable-length code with Exp-Golomb coding. Then only the information part in the coded stream is encrypted with a stream cipher.

This process is similar to table permutation. The main difference is that the permutation operation happens only in the codeword with the same length, and the key changes with the intra/inter-prediction mode. The decryption process is symmetric to the encryption one.

4.2 The CAVLC Encryption Algorithm (CEA)

This encryption algorithm combines encryption operation with Context-based Adaptive Variable Length Coding (CAVLC). That is, during CAVLC encoding, the DCT co-efficients are given as input into the CAVLC coder. The encoded sequence from CAVLC will consist of parameters such as the number of coefficients, trailing ones(T1) (coeff_token), the sign of each T1, the levels of the remaining non-zero coefficients, the total number of zeros before the last coefficient and each run of zeros are encoded respectively. Among them, the sign of each T1 and the levels of the remaining non-zero coefficients are more sensitive to the images understandability. In order to keep low cost and keep the code format unchanged, it is preferred to encrypt only few of the parameters. Thus, we propose to encrypt only the signs of T1 and the levels of the remaining non-zero coefficients while keep other parameters unchanged.

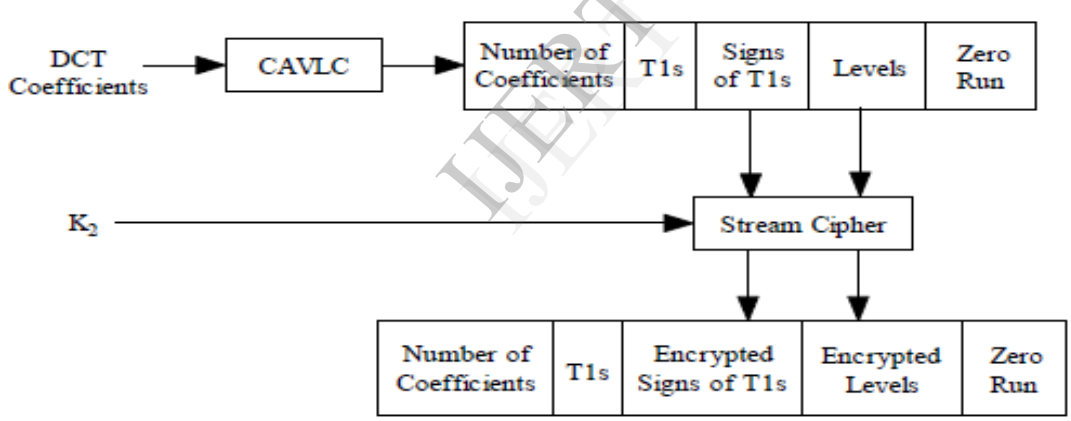

Fig. 2. The CEA encryption algorithm.

4.3 The Sign Encryption Algorithm (SEA)

In the proposed EGEA and CEA algorithms, a stream cipher is used to keep the code length and format compliance. Similarly, the intra-macroblocks ACs and inter-macroblocks' MVDs are also encrypted with a stream cipher followed with VLC coding, which makes it practical to realize real time stream media. Here, a stream cipher was presented.

Here, the encryption process is symmetric to the decryption process**.**

In this cipher, X(i) is the i-th plain-bit, and Y(i) is the encrypted one. The encryption mode is a kind of cipher text- feedback mode, in which, the i-th cipher-bit is the XOR- operation result among the i-th pseudo-random bit generated from the pseudo-random sequence generator, the i-th plain-bit to be encrypted and the pre-encrypted cipher-bit. The feedback register with the length of L is permuted before ciphertext feedback. The encryption process is under the control of k_1 and k_2.

5. Encryption during coding

5.1 VEA (VIDEO ENCRYPTION ALGORITM)

Rustam Rakhimov Igorevich, Hanmaro Yong (2010) has analyzed video encryption algorithm based on entropy coding.

Statistical analysis of compressed video is really different comparing with text or other data types, and after deeply learning that statistical properties of compressed video. Lintian Qiao and Klara Nardstedt triggered out their own developments of an efficient and secure VEA. MPEG has a more uniform distribution byte values. As a basic approach to achieve highest security level, there was considered sequence of I-frame and assumed to separate them to odd-numbered bytes and even-numbered bytes to form two new byte streams. As an operation over that streams was chosen

simple bitwise XOR operation and encryption function E.

5.2 USE (UNEQUAL SECURE ENCRYPTION)

USE is an unequal secure encryption and it's very useful for H.264/AVC video coding standard. The USE scheme encloses two parts: video data classification and unequal secure video data encryption. As an encryption algorithm used algorithm FLEX and XOR methods to reduce computational cost.

Data classification. All the data of total video were classified into two data partitions as important data partition and unimportant data partition. H.264/AVC has many new features that make this procedure easier to implement, and normally important data partition has smaller size than unimportant data. The importance of data is reckons up by how difficult to reconstruct a picture, after defining importance video data is parted into DPA (Data Partition A) and DPB (Data Partition B).

Unequal secure encryption. AES and FLEX encryption algorithms were used, regarding AES using to encrypt important data partition and FLEX use to encrypt unimportant data partitions. The computational cost of FLEX is only 1/5 of the AES encryption. Using this way achieved the scheme with highly secure and low computation cost.

In USE many data classification methods are used, the reason for that is primordially USE scheme was designed for H.264/AVC which has some of the new features.

Data partitioning (Extended profile) - This is a new feature of H.264/AVC making up a slices and placed them in three separate Data Partitions. First partition considered like important and contains the slice header and header data for MBs. Rest two partitions hold intra and inter coding MB's residual data.

FMO (Baseline Profile, Extended Profile) - is another new feature in H.264/AVC. It has an ability to partition the picture into slice groups (special regions). There are two kinds of partition modes in USE scheme, first is Region based FMO, here image can be partitioned into two slice groups: secret region and normal regions. Secret region can be selected by preprocessing tool and usually this region contains some recognized objects. Using this mode object-based encryption can be realized. The second partition mode is Mode Based FMO, and in this image also partitioned into two slices groups: Intra MBs and Inter MBs. Intra MBs contain important data's rather than Inter MBs, it means the Intra MBs must be highly secured than Inter MBs.

5.3 OPTIMIZED MULTIPLE HUFFMAN TABLES TECHNIQUE

Shaimaa A. El-said, Khalid F. A.Hussen (2011) proposed in-compression encryption using Optimized Multiple Huffman Table.It is a joint compression-encryption technique. OMHT coding technique used to both encode and encrypt video sequence is tested. It works under two scenarios. The first scenario is encoding video sequence frame by frame, and the other is encrypt the first frame and then compress and encrypt the motion vectors between successive frames.

The OMHT encryption technique OMHT process takes two parallel paths. It takes no additional time to add encryption to the compressed bitstream as both traditional and selective encryption techniques. It uses statistical-model-based compression to generate different tables from a training set of videos. This leads to increased compression efficiency and security. Multiple tables are generated for each one of videos types: A training dataset is constructed up as a number of selected frames of the same color palette as those to be encrypted. Each dataset is divided into N subsets each with equal number of frames. Random K frames are used to form single Huffman table. Concatenate all pixels values of all K images in single vector. Then, calculate the occurrence probability of each value, draw the Huffman tree, then generate Huffman table. Now we obtain D different (not fixed) Huffman tables for each dataset.

The number of tables that can be obtained from N frames is:

$$D = C_k^N$$

Preparing of the Optimized Tables

Following is the procedure of preparing and using the optimized multiple Huffman tables and how it is used to both encode and encrypt videos.

Step 1: Frames training set are divided into

datasets. Each dataset's frames have the same properties

Step 2: Each dataset contains N image

Step3: The input video's frames are compared to datasets to select the dataset that has the same properties

Step4: Randomly choose M subsets each subset contains K frames from the dataset. Concatenate all frames 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.

Step 5: Number the generated M table form 0 to M-1

Step 6: Generate a random vector P (the secret order) its length equal to the length of video frame under consideration. Each element value in P ranges from 0 to M-1.

Step 7: For the ith encountered symbol (coefficient to be encoded), use table Pi (mod n) to encode it

Step 8: Tables are saved at each decoder and the order (P) by which the tables are generated and used is kept secret.

Video Compression Using OMHT

OMHT model I processes the video sequence frames by frame and each frame is considered as a stand alone. Following is the procedure of compressing a video sequence .

1. Converting video sequence of any type into RGB frames.

2. Each nXm frame is converted into a single vector by cascading consecutive rows to form a single vector

3.This vector is exposed to a discrete cosine transformer to transform the spatial domain of frame's pixels into their frequency domain in which the frame energy concentrated in small number of coefficients.

4. The output of the DCT process is a vector that have the same length of the image (number of pixels in the frame), but with many values approximated to zeros. A selector is used to select the number of transmitted coefficients (Tc) that contains a specific percentage of the frame energy where as the other coefficients are canceled.

5. The selected coefficients are returned back into spatial domain using Inverse Discrete cosine transformation

6. The IDCT output is quantized using Range Mapping technique (RM) as discussed later in this section. This quantization technique is reversible; this means that the dequantized values can be turned back to their original values leading to no quantization losses

5.4 PERMUTATION OF THE HUFFMAN CODE

Another scheme proposed by Shi and Bhargava (1998) uses a permutation of the Huffman code word list as a secret key. During MEPG encoding (decoding), their encoder (decoder) uses the secret key instead of the standard Huffman code word list. Video frames decoded using the standard Huffman code word become incomprehensible. However, this method has a disadvantage in its difficulty in the generation of the encryption keys, resulting in having each generated key to be tested before using.

Zeng and Lei (1999) proposed a joint encryption and compression framework in which video data are scrambled in the frequency domain by employing selective bit scrambling, block shuffling and block rotation of the transformed coefficients and motion vectors. Their proposal performs encryption of the video data by scrambling the block coefficients and/or motion vectors. **A** cryptographic key will be used to control the scrambling. The scrambled coefficients and motion vectors are then subjected to video compression before they are transmitted over networks or stored in a storage device. (In other words, the scrambling is performed prior to compression [quantization and Huffman, run-length, arithmetic, embedded, or other entropy coding.])

At the decoder, the compressed video bitstream will first be decompressed (entropy decoding and dequantization). Authorized users will then use the same key to unscramble the decompressed coefficients before they are

subjected to inverse transformation, and unscramble the motion vectors before motion compensation.

5.5 SCRAMBLING OF THE HUFFMAN CODE WORDS

Mohan S. Kankanhalli and Teo Tian Guan (2002), the proposed scrambling of the Huffman code words which aims to reduce the overheads that are added to the encoding/decoding processes during encryption/decryption. In addition, it does not increase the compression ratio adversely. These twin aims are achieved by applying the techniques proposed in [161 on the Huffman tables that are used for Variable Length Coding (VLC). The resulting proposal is simple to implement; yet, as we will show, it provides reasonable security.

The Huffman encoding scheme can be considered as a *substitution algorithm*, where during encoding, it transforms (with a function **H)** the original video data **M** into variable length bit stream **B** with a key **k.** At the decoder end, a function **D** is used to obtain the original data from the bitstream with the use of the same key.

$$. M=D(H(M,K),K)$$

The Proposed Algorithm

The proposed scrambling algorithm consists of two methods:

1. Shuffling of the code words within each of the Huffman tables.
2. Random flipping of the last bit of the codeword

Shuffling of the Code Words for VLC. The code words used for VLC are shuffled within each Huffman table during the initialization of the encoding process. To minimize the impact on the compression rate, the code words for each of the tables are grouped into different subsets according to their bit length of code words. Different shuffling tables are generated for each Huffman table, increasing the difficulty of guessing the correct sequence. Similarly, when the correct values are obtained after re-shuffling during the decoder's initialization process, the entire video can be decoded with any further changes to the original program design. In addition, this scrambling method has a nice property that any shuffling permutation will not violate the prefix rule of Huffman coding. Therefore, there is no need for extra checks for the validity of the shuffling table to ensure that the decoder is able to decipher the concatenated code words. Random Bit Flipping of the Code Words for V L C. To increase the level of security, the last bit representing the code word is randomly flipped. However, to ensure the prefix rule is not validated, flipping of the last bit may require changing the bits of those affected code words.

5.6 ENCRYPTION WITH MULTIPLE STATISTICAL MODELS IN ENTROPY CODERS

Background and Motivation

The Huffman coder and the QM coder are the two most popular entropy coders in multimedia compression systems, and both have very simple statistical models. The statistical model of the Huffman coder is often a fixed-size non adaptive binary tree. During decoding a Huffman coded bit stream without any knowledge of the Huffman coding table would be tremendously difficult. This implies that an encryption system that uses the Huffman coding table as the *key* should be immune to cipher text only attacks, given that the table is not too small. However, it is obvious that this system is totally vulnerable to known-plaintext and chosen-plaintext attacks. Given a piece of the original data and its Huffman coded data, an attacker could determine many entries of the Huffman table by comparing the two data streams. When the length of the original data is long enough, the whole Huffman table could be resolved. Furthermore, in practical applications, the number of possible Huffman coding tables is often limited, and thus the key space is reduced.

Secure Huffman Coder With Multiple Coding Tables

Chung-Ping Wu and C.C Jay Kuo (2005) proposed encryption of multimedia data with

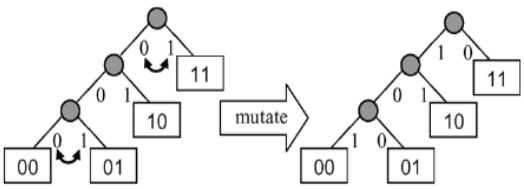

Fig. 3 a, b. Illustration of the Huffman tree mutation

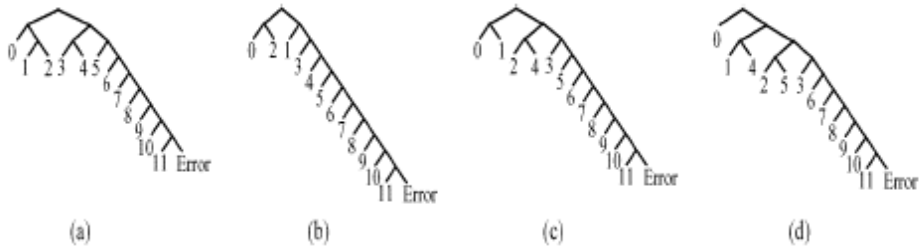

Fig. 4. Original Huffman coding tree for JPEG dc coefficient coding is given in (a), while three Huffman trees trained from three image sets are shown in (b), (c), and (d).

modification of entropy coding. Since the entropy coder is at the last stage of a media compression system, the successful use of selective encryption, the feasibility of integrating encryption into entropy coding are investigated in this section.

Huffman coding with a predefined fixed Huffman table is used in most modern multimedia compression systems, including MPEG video, MPEG audio and JPEG image compression. Since the coding of each symbol is only a simple table lookup process, it is the fastest compression algorithm. In this section, we first present and analyze the basic algorithm of the multiple Huffman tables (MHT) scheme. Then, two ways of enhancing the security of the MHT-encryption scheme are described.

Basic Algorithm: The input data stream is encoded using multiple Huffman coding tables. The content of these tables and the order that they are used are kept secret as the key for decryption. The basic algorithm can be described in the following three steps.

a) Choose m different Huffman coding tables. They are numbered from 0 to (m-1)
b) Generate a random vector P = (p_0, p_1, p_2, …….p_{n-1}), where each pi is a k-bit integer varying from 0 to (m-1), and k equals to [$\log_2 m$].
c) For the I th symbol in the original data stream, use table $p_{i(\text{mod } n)}$ to encode it.

It is important to generate the Huffman coding tables such that the compression ratio will not be affected. One way to achieve this goal is to generate each Huffman table from a different set of training images (or audio pieces). Although the resulting Huffman tables are different from each other, every Huffman table is equally "good" as long as each training set is a balanced representation of all images (or audio pieces) in the world. If two Huffman tables happen to be identical, we can simply pick up one of them. With this approach, thousands of different Huffman tables can be obtained. In Step 1 of the algorithm, we may randomly choose tables from the space of all available tables and send the indices as the key. If there is a total of 2^N published Huffman coding tables, each index would be N_t -bit long.

In the proposed system, we adopt an easier method to generate Huffman coding tables. Instead of training thousands of Huffman coding tables, we only train and obtained four different Huffman tables. Then, thousands of different tables can be de rived using a technique called Huffman tree mutation. in Fig. 3(a), a standard Huffman tree has every left-hand-side branch labeled "0" and every right-hand-side branch labeled "1". If we permute the label-pairs as indicated in Fig. 3(a), we will get a new Huffman tree as shown in Fig. 3(b). We call it the *Huffmantree mutation* process. Please note that each label-pair is attached to one inner node. For a Huffman table with t entries, its Huffman tree would have t leaves and (t-1) inner nodes and

label-pairs, which provides us the opportunity to make (t-1) decisions about whether to permute each label-pair. Therefore $^{2t-1}$Huffman trees can be derived. In order to produce a new Huffman tree, we first randomly generate a bit integer, then permute the label-pairs in the standard Huffman tree if the corresponding bit in the integer is zero. It should be noted that Huffman tree mutation has absolutely no influence on the coding efficiency in reference to the original Huffman tree.

The MHT algorithm can be illustrated with the following example. We choose the Huffman coder for JPEG dc coefficients because it has the right size for the demonstration purpose. Fig. 4(a) shows the original Huffman tree used in encoding JPEG dc coefficients. When JPEG standard was made, this tree was created using the statistics collected from a set of images, which is assumed to properly represent all of the images that the JPEG system will encode. Fig. 4(b)-(d) are three other Huffman trees that we trained with three independent image sets. The four Huffman trees in Fig. 4 are used as basic trees to derive others through Huffman tree mutation. Since each of these basic trees can be mutated into 2^{13-1} different Huffman trees, the total number of possible trees is $4*2^{13-1}=2^{14}$, i.e., N_t is equal to 14. Next, m of these possible Huffman trees are actually generated and used in the MHT algorithm as described above.

It is possible to replace the original Huffman tree in the standard with other trees and sometimes achieve a smaller com- pressed data size because "Huffman coding with fixed and pre-determined table" is inherently suboptimal in compression performance. The predetermined table is not tailored to the statistics of each image being compressed.

6. Conclusion

In this paper, we present three methods to improve the light weight video encryption algorithm for real time applications. We hope that further research in this area will result in more effective yet efficient designs. Different researchers have contributed their work to this field, and some of the algorithms are been explained here. Nowadays, such techniques are studied in wide application contexts, but the research is still challenging in the cryptographic community to design more efficient and secure schemes.

References.

1. T. Vino, E. Logashanmugam. A Model-based Multimedia Encryption Scheme for Real Time Videos. IEEE 2010.

2. L. Tang. Methods for encrypting and decrypting MPEG video data efficiently. In Proc. of ACM Multimedia, 1996.

3. Wenjun Zeng and Shawmin Lei. Efficient frequency domain selective scrambling of digital video. In Proc. of the IEEE Transactions on Multimedia, 2002, pp. 118–129.

4. Zhenyong Chen, Zhang Xiong, and Long Tang. A novel scrambling scheme for digital video encryption. In Proc.of Pacific-Rim Symposium on Image and Video Technology(PSIVT), 2006, pp. 997–1006.

5. Borko Furht, Daniel Socek, and Ahmet M. Eskicioglu. Fundamentals of multimedia encryption techniques. In Multimedia Security Handbook.

6. L. Qiao and Klara Nahrstedt. A new algorithm for MPEG video encryption. In Proc. of First International Conference on Imaging Science System and Technology, 1997, pp. 21–29.

7. Shi and Bharat Bhargava. A fast MPEG video encryption algorithm. In Proc. of ACM Multimedia, 1998, pp. 81–88.

8. Yan Li, Main Cai. H.264-Based Multiple Security Levels Net Video Encryption Scheme. IEEE International Conference on Electronic Computer Technology, 2009

9. Yajun Wang, Mian Cai. Design of a New Selective Video Encryption Scheme Based on H.264. International Conference on Computational Intelligence and Security, pp. 883-887, 2007.

10. Saranya.P, Varalakshmi.L.M. H.264 based Selective Video Encryption for Mobile Applications. International Journal of Computer Applications (0975-8887), Volume 17-No.4, March 2011.

11. Zheng Liu, Xue Li. Motion Vector Encryption in Multimedia Streaming. Proceedings of the 10th International Multimedia Modeling Conference. IEEE 2004.

12. Shiguo Lian, Zhongxuan Liu, Zhen Ren, and Haila Wang. Secure Advanced Video Coding Basedon Selective Encryption Algorithms. IEEE Transactions on Consumer Electronics, Vol. 52, No. 2, MAY 2006.

13. Rustam Rakhimov Igorevich, Hanmaro Yong, Dugki Min, Eunmi Choi. A Study on Multimedia Security Systems in Video Encryption. National IT Industry Promotion Agency-Korea.NIPA-2010-C 1090-1031-003.

14. Lintian Qiao and Klara Nahrstedt. *A New Algorithm for MPEG Video Encryption*. Department of Computer Science University of Illinois at Urbana-Champaign, 1304 West Springfield Avenue, Urbana IL 61801, U.S.A.

15. Shaimaa A. El-said, Khalid F. A. Hussein, Mohamed M. Fouad. Confidentiality and Privacy for Videos Storage and Transmission. International Journal of Advanced Science and Technology Vol.28, March, 2011.

16. Shi, C. and B. Bhargava. Light-weight MPEG Video Encryption Algorithm. In Proceedings of the International Multimedia Conference on Multimedia,(Multimedia98, Shaping The Future) January 23-25, 1998, pages 55-61, New Delhi, India. IETE, Tata Mcgraw-Hill Publishing Company.

17. W. Zeng and S. Lei. Efficient Frequency Domain Digital Video Scrambling for Content Access Control. Proceedings of the conference on ACM multimedia '99, 1999.

18. Mohan S. Kankanhalli, Teo Tian Guan. Compressed-Domain Scrambler/Descrambler for Digital Video. IEEE Transactions on Consumer Electronics, Vol. 48, No. 2, MAY 2002.

19. Chung-Ping Wu,C.-C. Jay Kuo. Design of Integrated Multimedia Compression and Encryption Systems. IEEE Transactions on Multimedia, Vol. 7, N0.5, October 2005.

20. D.Gillman, M.Mohtashemi, and Rivest. On breaking a Huffman code. IEEE Transactions on Information Theory, Vol. 42, no. 3, p.972, MAY 1996.