A Real-Time Video Encryption Exploiting the Distribution of the DCT coefficients

C. Narsimha Raju, Kannan Srinathan and C. V. Jawahar
International Institute of Information Technology
Hyderabad, India - 500032.
Email: {narsimha_raju@research., srinathan@, jawahar@}iiit.ac.in

***Abstract*—Most of the video encryptions algorithms considered in the literature significantly increase the video size independent of encryption algorithm employed. This is because these algorithms encrypt DCTs without considering their characteristics or relationship to the visual content. This adversely affects the transmission throughput. We propose a fast video encryption algorithm that exploits the statistics of the DCTs in the video data sets. Our algorithm reduces the video size significantly without any compromise in security. The proposed algorithm performs encryption followed by permutation of the DCTs. On an average, increase in video size is restricted to 23.41% of the original.**

***Index Terms*—Multimedia data security, MPEG codec, MPEG video encryption.**

I. Introduction

With active research in technologies related to network and multimedia, transfer of multimedia data over the network has become very easy, making the security and privacy issues in multimedia computing more and more important. Various encryption algorithms have been proposed in recent years as possible solutions for the protection of the multimedia data. Large volume of the multimedia data makes the encryption difficult using traditional encryption algorithms. Often, we need the encryption to be done in real-time.

The *naïve* approach for video encryption is to treat video data as text and encrypt it using standard encryption algorithms like AES (Advanced Encryption Standard) or DES (Data Encryption Standard). The basic problem with these encryption algorithms is that they have high encryption time. They also result in vast increase in size of the video, making them unsuitable for real-time applications like PAY-TV, Pay-Per View and Video On Demand (VOD) etc. A unique characteristic of video data is that, even though information rate is very high, information value is very low. Exploiting this fact, to decrease the encrypted video size and time, many selective encryption algorithms have been proposed which encrypt only selected parts of the data.

Meyer and Gadegast [1] have designed an encryption algorithm named SECMPEG which incorporates selective encryption and additional header information. In this encryption selected parts of the video data like Headers information, I frames, I-blocks in P and B frames are encrypted based on the security requirements. However, SECMPEG is not compatible with the standard MPEG and special encoder and decoder is needed to handle SECMPEG streams [13]. Qiao and Nahrstedt [2] proposed a special encryption algorithm named video encryption algorithm in which one half of the bit stream is XORed with the other half. The other half is then encrypted by standard encryption algorithm (DES). The speed of this algorithm is roughly twice the speed of *naïve* algorithm, but that is arguably still the large amount of computation for high quality real-time video applications that have high bit rates [13].

Some of the other encryption algorithms are based on scrambling the DCT coefficients. Tang's [3] scrambling method is based on embedding the encryption into the MPEG compression process. The basic idea is to use a random permutation list to replace the zig-zag order of the DCT coefficients of a block to a 1×64 vector. However, changing the zig-zag order to a random order, results in an increase of video size, about 25% to 60% which is not tolerable. Zeng and Lie [4] extended Tang permutation range from block to segment, with each segment consisting of several macroblocks. Within each segment, DCT coefficients of the same frequency band are randomly shuffled within the same band. Chen, *et. al* [5] further modified this idea by extending the permutation range from a segment to a frame. Within a frame, DCT coefficients are divided into 64 groups according to their positions in 8×8 size blocks, and then scrambled inside each group. Apart from shuffling of the I frames, they also permuted the motion vectors of P and B frames.

In order to meet the real-time requirements, *light-weight encryption* algorithms were proposed. These encryption algorithms are fast. They add less overhead on the codec. Shi, *et. al* [6] proposed a light-weight encryption algorithm named Video Encryption Algorithm (VEA). It uses simple XOR of sign bits of the DCT coefficients of an I frame using a secret m-bit binary key. The algorithm was extended as Modified Video Encryption Algorithm (MVEA) [7] wherein motion vectors of P and B frames are also encrypted along with I frames.

In this paper, we propose a novel selective cum light-weight encryption algorithm. This algorithm is fast, making it suitable for real-time applications, yet possessing practically acceptable levels of security without much overhead on the MPEG encoding and decoding process. The basic idea of this algorithm is to perform encryption followed by permutation of the DC and AC coefficients based on the statistical properties of the DCT coefficients. This results in a minimal increase of the video size. Our algorithm takes an average encryption

time of 7.2 milliseconds per frame, making it ideal for real-time encryption.

The rest of the paper is organized as follows. Section 2 gives the background of MPEG. The details of the proposed encryption algorithm are presented in Section 3. Experiments and security analysis are given in Section 4. Finally, we conclude in Section 5.

II. Background

MPEG [9] is the acronym for Moving Picture Expert Group. Its standards including MPEG-1, MPEG-2 and MPEG-4 are widely used for multimedia storage and communication. These scheme are based on the principle of motion compensated prediction and block-based transform coding. Often Discrete Cosine Transform (DCT) is used for the compression of these videos.

A MPEG video is composed of a sequence of Group Of Pictures (GOPs). Each GOP consists of I, P and B frames. I frames are called intra-coded frames. These frames are further split into non-overlapping blocks (intra-coded) of 8×8 pixels which are compressed using DCT followed by quantization, zig-zag scan, run-length coding and entropy coding.

The P and B frames are forward predictive coded frames and bi-directional predictive coded frames respectively. These are subjected to motion compensation by subtracting a motion compensation prediction. The residual prediction error signal frames are also split into non-overlapping blocks (inter-coded blocks) of 8×8 pixels which are compressed in the same way as the blocks of intra-frames. Sometimes, P and B frames also have some intra-coded blocks when better efficiency will be obtained using intra-coded compression [5]. These intra-coded blocks are called I-blocks in P and B frames. Figure 1 shows the block diagram of the MPEG video coding scheme.

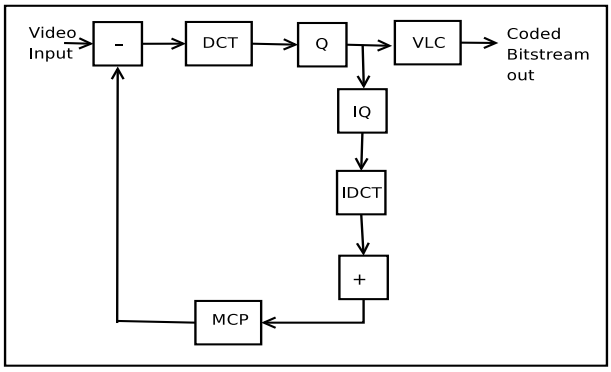

Fig. 1. MPEG Video Coding Scheme

In 8×8 DCT transform coding, the 64 transformed coefficients are zig-zag ordered such that coefficients are arranged approximately in the order of increasing frequency. The DCT transform coefficients can be classified into two groups namely, DC and AC coefficients. The DC coefficient determines the average brightness in a block. All other coefficients describe the variation around this DC value and these are referred to as AC coefficients.

III. Encryption Algorithm

In a particular DCT block, most of the energy is concentrated in the DC and very few AC coefficients. According to Liu, *et. al* [8] and Meyer, *et. al* [1], encryption of DC and first three to nine AC coefficients is sufficient to hide the details of a given video block. In our encryption algorithm, we consider DC and first nine AC coefficients for encryption. In order to encrypt these coefficients efficiently, we exploited the distribution of DCT coefficients in the video data. The proposed algorithm is selective cum light-weight encryption algorithm consisting of two steps. In the first step of encryption, the DCT coefficients are XORed based on the statistical analysis of the DCT coefficients. In the next step, permutation of the DCT coefficients is done. Before we go into the finer details of the encryption scheme, we present the statistical analysis of the video data in the next subsection.

A. Statistical analysis of the video data

The DCT coefficients obtained in the MPEG compression are typically quantized and encoded using Huffman coding. The length of the Huffman code-word depends on the DCT coefficients. Smaller coefficients are encoded with short code-words and large code-words are used to represent large coefficients. Table 1 shows the length of the Huffman encoding for different ranges of the DC and AC coefficients.

Range	DC Coefficient Length	AC Coefficient Length
0	3	N/A
-1, 1	4	4
-3, -2, 2, 3	5	5
-7, . . . -4, 4,. . . , 7	5	5
-15, . . . -8, 8, . . . ,15	7	7
-31, . . . -16, 16, . . . , 31	8	8
-63, . . . -32, 32, . . . , 64	10	10
-127, . . . -64, 64, . . . , 128	12	12
-255, . . . -128, 128, . . . , 255	14	14

TABLE I
Length of Huffman Codewords

Statistical analysis is done on the DCT coefficients after the Quantization (Q) operation and before Variable Length Coding (VLC) (as shown in Figure 1). To estimate various properties of the DC and AC coefficients, a large set of videos taken from different sources have been analyzed. The selected videos consist of both bench-marked and fresh sources, comprising of movie-clips, home videos, videos collected from Internet as well as standard test videos like Foreman, Mother and Table Tennis videos. The set has been chosen such that they exhibit a variety of motion characteristics with objects in the video content varying at different speeds. We found some really interesting behavior in the frequency distribution of DCT coefficients in various video blocks.

For this particular set of fifty videos considered for the experiments, the plot in Figure 2 shows the percentage of DC coefficients that fall in the range of $0 - 50$ and $50 - 255$ respectively. It can be observed from the plot that in most of

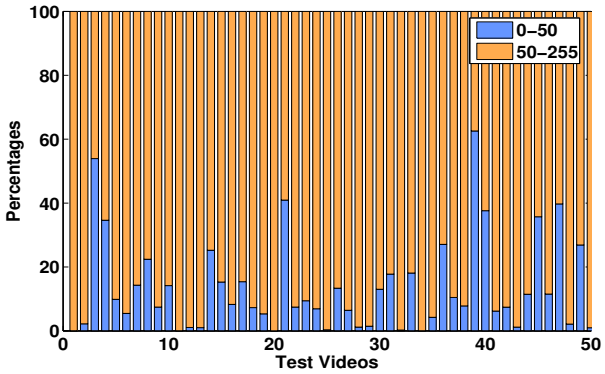

Fig. 2. Percentage Variation Plot of DC Coefficients

the videos, a high percentage (86.53%) of DC values lie in the range of $50-255$. This is because DC coefficient represents the mean value of the given image block and carries most of the energy in a video block.

Similar analysis was done on the AC coefficients. More specifically we limit the analysis on ACs to first nine coefficients (as only nine AC coefficients are used for encryption). Since the variation of the ACs is completely different from the DC, we used the range $0-30$ and $30-255$. It can be observed from Figure 3 that most of the stack bar (min. 96.94%) is occupied by $0-30$ range for each of the top nine AC coefficients than that of $30-255$. This is due to the fact that AC describe the variation around the DC. From this analysis, we conclude that most of the AC coefficients lie in the range $0-30$ and hence these coefficients need smaller length Huffman codewords as shown in Table 1. Figure 3 is constructed by discarding the sign of the DCT coefficients.

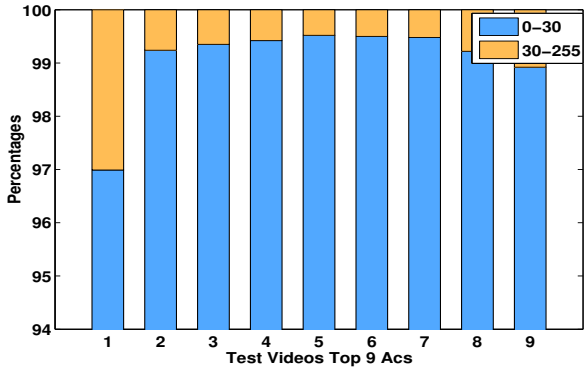

Fig. 3. Percentage Variation Plot of First Nine AC Coefficients

Most of the video encryption algorithms do not consider this observation and use random keys to encrypt all the DC and AC coefficients. Ignoring this aspect leads to very high increase in the video size because encryption of the DCT coefficients by random keys results in encrypted coefficients to lie in the range of 0 to 255. Orthogonal to these techniques, we propose a technique which considers the properties of the DCT coefficients and performs encryption accordingly, resulting in minimal increase in video size without any loss in security level. This technique is also computationally efficient, requiring less encryption time. This statistical analysis motivated us to design an encryption algorithm for the video data.

B. Proposed algorithm

Algorithm 1 briefly explains the proposed method of encryption. Let us assume that a multimedia video data (V) after quantization consists of the n 8×8 macroblocks (mb) denoted by

$$V = \{mb_1, mb_2, mb_3 \dots mb_n\}$$

Each mb_i consists of atmost 64 DCT coefficients denoted by

$$mb_i = \{DC_i, AC_{i,1}, AC_{i,2} \dots AC_{i,63}\}$$

Algorithm 1 Encryption Algorithm

Step 1: Generate a set of random numbers using PRNG which takes the seed (K_s) as an input.
Step 2: K_s is used to generate 64 different permutation tables using second PRNG.
Step 3:
for each and every frame (V) **do**
  Step 3.1: Perform the XOR operation of atmost 10 DCT coefficients of each block (mb_i) using values generated from first PRNG.
end for
Step 4: Group the DCT coefficients of a frame according to their position in a 8×8 block.
Step 5: Shuffle the DCT coefficients of a frame using the 64 different permutation lists generated from second PRNG.

We apply XOR of the DC and first nine AC coefficients by a Pseudo-Random Number Generator (PRNG). The generated values from PRNG for DC are between 50 to 255 and for AC values are between 0 to 30. The ranges are chosen such that the result of encryption lie in the range of common values of the video data. Thus, increase in video size is significantly reduced. We then permute the encrypted coefficients of the frame. In this permutation step, the DCT coefficients of a frame are divided into 64 groups according to their positions in 8×8 size blocks, and scrambled within a group [5]. The grouping of the DCT coefficients is shown in the Figure 4. The shuffling list is generated by another PRNG.

The first PRNG module takes seed (K_s) as input and generates a sequence of random numbers. These generated values are used for performing the XOR operation. The second PRNG takes the same seed as input and generates 64 permutation tables. These tables are then used to permute the encrypted DCT coefficients.

The security of the proposed algorithm depends on the seed to the PRNG. In order to protect the seed, it can be encrypted by the conventional public-key cryptosystem like RSA and sent to the receiver. In order to make the algorithm more

(a) Athletics(171) (b) Horse(166) (c) Penguin(51) (d) Pond(75)

(e) Rat(89) (f) Lab(245) (g) House(1977) (h) Table Tennis(13)

Fig. 5. Example Test Videos along with Frame Numbers

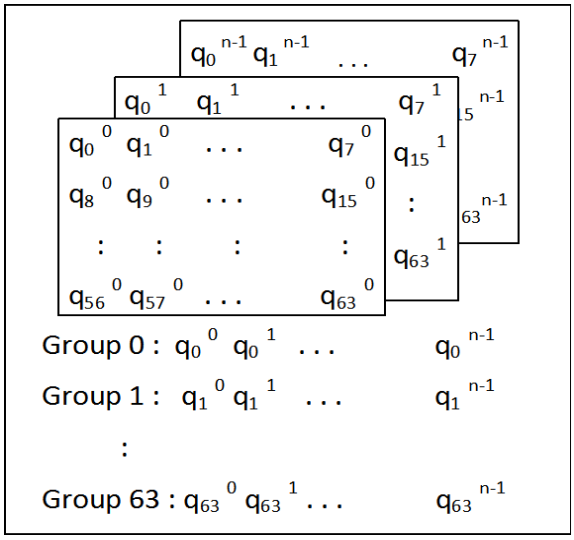

Fig. 4. Grouping of the Quantized DCT Coefficients

secure, key (K_s) is renewed after every periodic interval. At the receiver end, the authorized user decrypts using his RSA private key to get the key (K_s), generates the random numbers and 64 permutation tables with the help of it and performs the inverse permutation followed with XOR operation for getting the original content of the video data.

IV. Experiments and Security Analysis

A. Experimental Results

The proposed algorithm is implemented on a dual-core processor of 1.0 GHz each, with 1 GB of RAM. The algorithm is coded using Cornell University DALI [10] library, which supports coding of MPEG-1 and MPEG-2 videos. The test videos include Table Tennis, Pond, Penguin and some other general videos each with different motion characteristics.

Apart from the security (which is explained in the next subsection), an encryption algorithm for real-time multimedia data should possess two characteristics. First, compression rate of the video should not be decreased making it difficult to transmit over the network. Second, as the encryption time adds delay in transmission, it should not be high. Our proposed algorithm possesses both these characteristics with no extra burden on the codec.

The proposed algorithm encrypts atmost 10 DCT coefficients (1 DC and 9 ACs) for each macroblock. The test videos are shown in Figure 5 along with the frame number. The results of XORing and scramble the test videos are shown in Figure 6. Visual details in the video are almost lost after the first step of encryption, but still a very low-quality image can be seen. The details are completely lost after scrambling the XORed videos.

We also experimented by encrypting all the DCT coefficients of a macroblock and observed the increase in video size and degradation in the quality of the video. The increase in the video size was found to be 32.40% with hiding most of the video details. We then proceeded encrypting only 10 DCT coefficients, we found that increase in video size was less with equivalent loss in video quality. Since the chrominance component deals with the color information, it does not add any advantage to the proposed encryption technique. However, it causes increase in video size and encryption time. Therefore only the luminance component of the video data is encrypted and satisfactory results were obtained.

When there are many motive objects and/or scenes in a video file, corresponding P/B frames have intra coded blocks, leaking partial image information. In such cases, to achieve higher security, we should encrypt the DCT coefficients of P/B

(a) Athletics(171) (b) Horse(166) (c) Penguin(51) (d) Pond(75)

(e) Rat(89) (f) Lab(245) (g) House(1977) (h) Table Tennis(13)

Fig. 6. Frames from Encrypted Test Videos

frames also. Theoretically, absolute security is achieved when intra-coded blocks of P/B frames are also encrypted along with I frames. However, such an encryption increases time overhead intolerably. Therefore, as a substitute, one can encrypt motion vectors of P/B frames because we can achieve same encryption efficiency with much less overhead. Sometimes the leakage of partial image information might help in attracting a non-paid customer, creating interest in buying the video. Many of the real-time applications do not need such high security [14], but if one desires, he can XOR the motion vectors of P/B frames using the first PRNG. First PRNG is used because the variation of motion vectors is -64 to $+64$ followed by permutation using second PRNG.

In order to study the effectiveness of the proposed technique by considering the two metrics which are stated above, we conducted experiments by using other encryption algorithms like XOR, XOR with scramble [12] and XOR of 10 DCT coefficients (without statistical analysis). In all these encryptions, we encrypted only the luminance component and comparsions are done. Figure 7 shows the overhead in the video size of the corresponding encryption algorithms. The average increase in video size by our method is far less than complete XOR encryption (62.2%). Even if we restrict encryption to 10 DCT coefficients using XOR , the increase in video size is found to be 41.77%. However by exploiting the distribution of the DCT coefficients further, we obtain significant reduction in video size. On an average the overhead in the video size using the proposed method is only 23.41%. It can be inferred that DES and AES encryptions incur similar overhead in size as that of XOR because even they do not consider the characteristics of the DCT coefficients.

The average encryption time of our technique is 7.2 milliseconds per frame, which is much less than encryption time using AES, DES [1], RC5, XOR followed by scrambling [12].

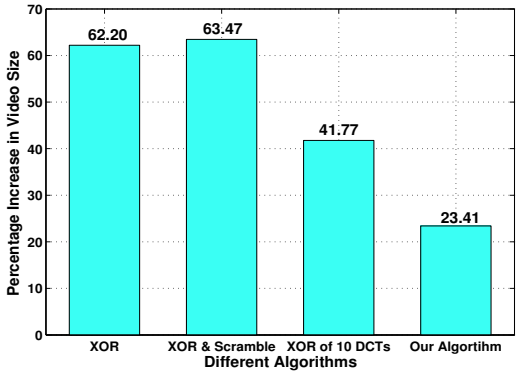

Fig. 7. Encryption Overhead on the Video Size

The encryption time is less because we encrypt only 10 DCT coefficients per block. On the other hand, the encryption time taken by AES, DES [1] and RC5 are high because they are standard cryptographic algorithms designed for textual data encryption.

For few test videos, Table 2 tabulates the results of our proposed technique along with other methods of encryption. It can be seen from the table our method of encryption adds less overhead on the codec.

B. Security Analysis

For multimedia data, an encryption algorithm is said to be *computationally* secure if the cost to break encryption by an cryptanalyst needs more investment than buying the key itself. There are multiple ways of attacking the encrypted videos. The security of the proposed algorithm is checked against most common ways of attacking the videos like ciphertext-only attack and known-plaintext attack.

Name	Original size (Kb)	Total Number of Frames	Total I Frames	XOR Encryption	XOR and Scramble	XOR of 10 DCTs	Proposed Method of Encryption
Athletics	280	184	12	464	464	428	360
Horse	1968	801	54	2696	2700	2644	2340
penguin	400	86	8	648	650	592	496
Pond	5144	611	61	7256	7280	6352	5784
Rat	1592	315	27	2164	2164	2036	1840
Lab	2228	635	43	3088	3088	1760	2484
House	6612	4320	360	10248	10260	10052	8680
Table Tennis	1224	150	26	2948	2956	1840	1544

TABLE II
RESULTS OF COMPARSION WITH VARIOUS METHODS OF ENCRYPTION

1) Ciphertext-only Attack: In the case of ciphertext-only attack the attacker has the encrypted video and the encryption algorithm. This is the most difficult attack since the cryptanalyst does not know anything about the plaintext (original video). For a frame, the attacker need to try all the combinations in order to get back the original video. For a video frame of size 384×288, the number of luminance macroblocks would be 48×36 and assuming that n_y be the number of non-zero luminance components per frame, the computational complexity of breaking the first step would be $255^{48\times36\times10}$ (Here 10 specify the number of XOR operations per block). The computational cost of breaking the permutation step would be $\pi_{i=1}^{64} n_{y_i}!$, where $n_{y_i} \leq 48 \times 36$. Hence the overall cost of XOR and permutation makes breaking the video file practically infeasible, making our technique is robust to ciphertext-only attack.

2) Known-plaintext Attack: In the case of known-plaintext attack, the unauthorized user has the original video, the corresponding encrypted video and the encryption algorithm. Though PRNG is less secure to known-plaintext attacks, our algorithm provides computational security. Moreover to strengthen the ability against the known-plaintext attack, renewing of the key has been done at periodic intervals. Apart from that, we do not encrypt initial frames to avoid known-plaintext attack from predictable frames such as MGM roaring lion sequences. Hence it is secure against known-plaintext attack also.

V. CONCLUSION

For real-time multimedia applications, light-weight encryption algorithms are attractive. In this paper we presented a fast novel selective cum light-weight encryption algorithm for the security of multimedia data. Even though the security of the proposed algorithm depends on the security of pseudo-random number generator, the cost to break the security of the proposed algorithm is far higher than buying the video. Theoretical analysis and experimental results show that the proposed encryption scheme provides computational security, high speed and has less overhead on the compressed video stream. All these advantages make it highly suitable for real-time multimedia transfer. Future directions of work include extending the analysis on the DCT coefficients and encrypting them using standard encryption algorithms like AES, DES, RC5. We also look for possible methods to further reduce the size of the encrypted video.

REFERENCES

[1] J. Meyer and F. Gadegast, *Security Mechanisms for Multimedia Data with the Example MPEG-1 Video, In Web, 1995.*

[2] L. Qiao and Klara Nahrstedt, *A New Algorithm for MPEG Video Encryption, In Proc. of First International Conference on Imaging Science System and Technology, pp 21–29, 1997.*

[3] L. Tang, *Methods for Encrypting and Decrypting MPEG Video Data Efficiently, In Proc. of ACM Multimedia, pp 219-229, 1994.*

[4] Wenjun Zeng and Shawmin Lei, *Efficient Frequency Domain Selective Scrambling of Digital Video, In Proc. of the IEEE Transactions on Multimedia, pp 118-129, 2002.*

[5] 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), pp 997-1006, 2006.*

[6] C. Shi, S. Wang, and Bharat Bhargava, *MPEG Video Encryption in Real-time Using Secret Key Cryptography, In Proc. of International Conference on Parallel and Distributed Processing Techniques and Applications, 1999.*

[7] C. Shi and Bharat Bhargava, *A Fast MPEG Video Encryption Algorithm, In Proc. of ACM Multimedia, pp 81-88, 1998.*

[8] Zheng Liu, Xue Li, and Zhaoyang Dong, *Enhancing Security of Frequency Domain Video Encryption, In Proc. of ACM Multimedia, pp 304–307, 2004.*

[9] P. N. Tudor, *MPEG-2 Video Compression, Electronics and Communication Engineering Journal, 1995.*

[10] Wei Tsang, Brian Smith, Sugata Mukhopadhyay, Haye Hsi Chan, Steve Weiss, Matthew Chiu, and Jiesang Song, *The DALI Multimedia Software Library, IEEE Second Workshop on Multimedia Signal Processing, 1999.*

[11] T. B. Maples and G. A. Spanos, *Performance Study of a Selective Encryption Scheme for the Security of Networked, Real-time Video, In Proc. of Fourth International Workshop on Multimedia Software Development, 1995.*

[12] L. S. Choon, A. Samsudin, and R. Budiarto, *Light-weight and Cost-effective MPEG Video Encryption, In Proc. of Information and Communication Technologies: From Theory to Applications (ICTTA), pp 525-526, 2004.*

[13] Borko Furht and Darko Kirovski, *Multimedia Encryption Techniques, Multimedia Security Handbook, 2004.*

[14] B. Macq and J. Quisquater, *Cryptology for Digital TV Broadcasting. Proceedings of IEEE, pp 944-957, 1995.*