Jurnal Teknologi

Full Paper

IMPROVE THE PERFORMANCE OF MPEG VIDEO ENCRYPTION ALGORITHM USING MODIFIED RC4 ALGORITHM BASEDON CHAOTIC MAP

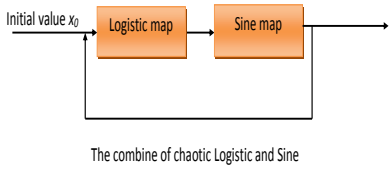
Ali Abdulgader*, Mahamod Ismail, Nasharuddin Zainal

[a]Department of Electrical, Electronic, & Systems Engineering, Universiti Kebangsaan Malaysia,43600 UKM Bangi, Selangor Darul Ehsan, Malaysia

*Corresponding author aliabdulgader2004@yahoo.com

Article history
Received
5 November 2014
Received in revised form
2 February 2015
Accepted
15 June 2015

Graphical abstract



Abstract

With rapid growth in communication techniques and multimedia application, security is becoming more important for multimedia data storage or transmission. Traditional encryption algorithm such as Advance Encryption Standard (AES) and Data Encryption Standard (DES) are not suitable for video data because it caused large computational overhead and reduce the MPEG compression efficiency. The encryption algorithm that has less computation overhead is needed. The proposed algorithm combines encryption process with MPEG video compression. Some modifications of the RC4 algorithm were proposed in this paper to enhance the performance of video encryption algorithm in terms of encryption time, compression ratio and to provide accepted level of the security. The modification of the RC4 algorithm depends on the maximum value of the plain text, chaotic map and circular shift operation. This modification will reduce the number of iteration in the Key scheduling Algorithm (KSA) and make the Pseudo Random Generator Algorithm (PRGA) depend on initial condition of chaotic maps to provide the strong shuffling inside the PRGA. The random rotation technique based on chaotic map is proposed to increase security of MPEG video. This technique depends on the value generated from chaotic map to rotate the Y blocks in I frame. The experimental results indicate that the proposed method provided better encryption times and provides sufficient level of the security. Thus the proposed method is suitable to protect the MPEG video with minimizing the overhead.

Keywords: RC4 algorithm, MPEG video encryption, chaotic map

Abstrak

Dengan pertumbuhan pesat di kalangan teknik komunikasi dan aplikasi multimedia, keselamatan menjadi semakin penting untuk penyimpanan atau penghantaran data multimedia. Algoritma penyulitan tradisional seperti Advance Encryption Standard (AES) dan Standard Penyulitan Data (DES) tidak sesuai untuk data video kerana ia menyebabkan overhed pengkomputeran besar dan mengurangkan kecekapan mampatan MPEG. Suatu algoritma penyulitan yang mempunyai kurang overhed pengiraan diperlukan. Algoritma yang dicadangkan menggabungkan proses penyulitan dengan mampatan video MPEG. Beberapa pengubahsuaian algoritma RC4 telah dicadangkan di dalam kertas ini untuk meningkatkan prestasi algoritma penyulitan video dari segi masa penyulitan, nisbah mampatan dan untuk menyediakan tahap keselamatan yang boleh diterima. Pengubahsuaian Algoritma RC4 bergantung kepada nilai maksimum teks yang jelas, peta huru-hara dan operasi peralihan bulat. Pengubahsuaian ini akan mengurangkan bilangan lelaran dalam Algoritma penjadualan Utama (KSA) dan membuat Algoritma Penjana Rawak Pseudo (PRGA) bergantung kepada keadaan awal peta huru-hara untuk menyediakan pengocokan yang kukuh di dalam PRGA tersebut. Teknik putaran rawak berdasarkan peta huru-hara dicadangkan untuk meningkatkan keselamatan video MPEG. Teknik ini bergantung kepada nilai yang dijana dari peta huru-hara untuk memutar blok-blok Y di dalam bingkai

I. Keputusan eksperimen menunjukkan bahawa kaedah yang dicadangkan memberi masa penyulitan yang lebih baik dan memberi tahap keselamatan yang mencukupi. Justeru itu kaedah yang dicadangkan adalah sesuai untuk melindungi video MPEG dan sekaligus meminimumkan overhed.

Kata kunci: Algoritma RC4, penyulitan video MPEG peta huru-hara

1.0 INTRODUCTION

Protecting multimedia data from attacker and illegal copying takes attention of many researchers in the recent years, especially with rapid development in communication networks and multimedia application. Joint multimedia encryption and compression schemes come to provide encryption at low overheads and without changing the structure of multimedia content [1]. The encryption algorithms should not reduce the MPEG compression ratio and should provide acceptable level of security and fast enough to satisfy real time constraints. Conventional encryption algorithm such as Advance Encryption Standard (AES) and Data Encryption Standard (DES) are not suitable to encrypt video data because it caused large computational overhead and reduce the MPEG compression efficiency. The encryption algorithm that has less computation overhead is needed [2].

The paper is arranged as follows. Previous work in MPEG video encryption is illustrated in Section 2. A brief introduction to RC4, proposed modified RC4and encryption algorithm are stated in Section 3. Experimental results are discussed in Section 4. The paper concludes in Section 5.

2.0 RELATED WORK

Many encryption techniques have been proposed to encrypt MPEG video. One of these techniques is combining encryption techniques with MPEG video compression and it is called joint compression and encryption algorithms. The output of transformation stages Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT)) or motion estimation stage during the compression process was exploited by these encryption techniques for the encryption. Shi and Bhargava [3] proposed method that exploits the DCT coefficients in every block for encryption. In their method, the binary key was generated and it used to encrypt the DCT coefficients and motion vectors encrypted by using XOR operation. This approach is quite simple and its dose not change the bitstream format but it is weak when the attacker gets some plaintexts of the encrypted video data.

Block Shuffle is the selective encryption method proposed by Tang and lei [4]. During this method, the 64 byte permutation table that was generated and applied to the 8 × 8 blocks rather than zigzag order of MPEG to shuffle the DCT coefficients in blocks. Replacing the original zigzag order with a 64 byte random sequence in this method caused large amount of bit overhead after the entropy coding stage. Experimental results show that the bit overhead becomes 108% for I-frame encryption and 55% for the entire encryption of sequence "Carphone," respectively.

Zeng and Lei [5] propose another technique that joint encryption and compression process. This method uses scramble technique to encrypt MPEG video. The DCT coefficients are divided into sub bands such as a block, slice or frame which consists of a number of macroblocks. The scramble technique changes the position of DCT coefficients in a sub band according to a permutation table. The proposed method is very simple to implement, makes the video become distorted, and provides considerable levels of security. The bit overhead caused by shuffle the sub band of slice data about 9.1% and 19.8% when processing I-frame and all the sequence, respectively.

In MPEG compression at the entropy coding stage, the DCT coefficients of each 8 by 8 block are encoded into codewords. The codewords consists of length code and binary values of DCT coefficients. The length code denotes the number of bit which required representing the value of the DCT coefficient. It is variable length coded according to tables that used with MPEG compression [6] [7]. The DCT components in each block are mapped to a new value by conventional encryption schemes, coefficients of small (in magnitude) values are mapped to larger values. Hence, without redesign the Huffman table requires more bits to code the encrypted frame when compared to coding frame without encryption. This motivates us to modify one of standard encryption algorithm (RC4 algorithm) to suppress bitstream size increment, and to avoid unauthorized viewing of MPEG video frames.

3.0 INITIAL RC4 ALGORITHM AND MODIFIED VERSION

3.1 The RC4 Algorithm

The RC4 algorithm is one of the important cryptosystem algorithms that widely used in the

network communication to protect any kind of the data. The RC4 is a stream cipher that used to achieve the confidentiality in encryption systems and it has advantage such as less time utilization during encryption and decryption. Although the RC4 algorithms have several weaknesses, but still many applications make use for it, especially the applications need fast encryption schemes. The encryption and decryption stages of the RC4 algorithm consist of Key scheduling Algorithm (KSA) and Pseudo Random Generator Algorithm (PRGA) respectively [8]-[9]. During the KSA, the randomization of the state matrix of size 256 are based on a variable-length key from 1 to 256 bytes (8 to 2048 bits) and two 8-bit index-pointers (denoted "*i*" and "*j*") that used to swap bytes in state matrix. In PRGA, at each loop during the encryption or decryption, a one-byte stream key is generated by using the 256-byte permutation state S and two byte indices i and j and then it is XORed with one-byte of the plaintext/ cipher text. The permutation state S as well as the two byte indices i and j are updated at each encryption and decryption loop as shown in Table 1.

Table 1.The two stage of the RC4 algorithm.

KSA	PRGA

/* Initialization state_matrix S*/
for i = 0 **to** 255 **do** {
S [i] = i;
}
end
/* Initial Permutation of state_matrix S*/
j = 0;
for i = 0 **to** 255 **do** {
j = (j + S [i] + S [i mod key length]) mod 256;
Swap (S [i], S [j]);
}
end
	/* Stream Generation */
	i, j = 0;
	while (true)
	i = (i + 1) mod 256;
	j = (j + S [i]) mod 256;
	Swap (S [i], S [j]);
	output=S[S[i] + S [j]] mod 256]]
	end

3.2 The Proposed Modified RC4 Algorithm

The idea of modifying the RC4 algorithm is to enhance the performance of video encryption algorithm in terms of encryption time, compression ratio and security and also to make it flexible encryption algorithm depending on maximum values of image intensity or maximum values of DCT components in each video frame instate of fixed size of state matrix (256 byte). The proposed method to modify RC4 is based on the maximum value of the plain text and chaotic map. This modification will reduce the number of round in the KSA and make the PRGA depend on initial condition of chaotic maps to provide the strong shuffling inside the PRGA. The two traditional 1D chaotic maps which will be used inside the KSA and PRGA in RC4 algorithm are:

3.2.1 Logistic Map

Logistic map was developed in 1974 by Mitchell Feigenbaum [10]-[11]. Mathematically, the logistic map is defined as in (1):

$$x_{n+1} = \lambda \times x_n \times (1 - x_n) \quad (1)$$

where λ is a parameter and $\lambda \in$ [0, 4], n is number of iteration , is the initial value within range [0, 1]. The chaotic behavior of logistic map can be achieved when parameter λ>3.57.

3.2.2 Sine Map

The sine map [12] [13] is another chaotic map that is used in the proposed method to increase randomly in PRGA. Mathematically it is define as in (2):

$$x_{n+1} = r \times sin(\pi \times x_n) \quad (2)$$

where r is a control parameter between 0 and 1, n is the number of iteration and x_0 is the initial condition when n=0 and the input/output with a range of [0, 1]. The chaotic behaviors of sine map achieved when the parameter $r \in$ [0.867, 1]. The combination of logistic and sine maps are to generated another chaotic map. This map has better chaotic properties because it is acquired chaotic behavior for each of the two maps. The combination of Logistic and Sine maps generated sequences by using feedback process in which the output of one map is used as the input of the other map as shown in Figure 1. Mathematically, the logistic-sine map is defined as in (3):

$$x_{n+1} = r \times sin(\pi\lambda \times x_n \times (1 - x_n)) \quad (3)$$

where *r* and λ are parameters, and $r \in$ [0, 1], $\lambda \in [0,4]$. The logistic-sine map has a large chaotic range of both parameters *a* and *r*, especially on the parameter a.

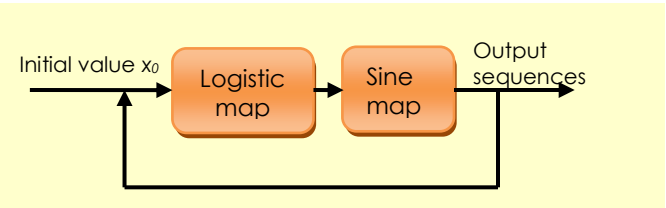


Figure 1 The combination of logistic and sine maps

The proposed modified RC4 algorithm is divided into two parts KSA and PRGA. During the KSA, the RC4 algorithm is based on fixed size of state matrix (256 byte) while in proposed method the state matrix is variable size depend on the maximum value of the plain text, the elements of state array S are variable

from 0 to the maximum data value. The output of KSA is random state array S from 0 to maximum data value. In PRGA, the elements of S depending on chaotic maps (Logistic-Sine maps) and circular shift operation to increase the randomness on generating the j value that uses as index location pointer. In this algorithm, the initial condition λ and state matrix S from the KSA are used as input in the PRGA as shown in Table 2. The elements of state matrix S are swapped by three pointers: i, j and k instead of two in the original RC4 algorithm. The output of the PRGA is the two bytes keystream per iteration which will be XORed with plaintext to get cipher text.

Table 2 The two stages of the modified RC4 algorithm

KSA	PRGA
Input: key, maximum data value Output: key stream S	Input: key sequences S , chaotic parameters (x0) Output : key streams

/* Initialization state_matrix S*/
for i = 0 ***to*** max data ***do*** {
S [i] = i;
}
end
/* Permutation of state_matrix S*/
j = 0;
for i = 0 ***to*** max ***do*** {
j = (j + S [i] + S [i mod key length]) mod (max data+1);
Swap (S [i], S [j]);
}
end
	/* Stream Generation */
	i, j = 0;
	for i = 0 ***to*** Length(data)/2 ***do*** {
	$x_{n+1} = 0.98 \times sin(\pi \times x_n(1-x_n))$
	$K = floor\left(mod\left(x_n \times 10^4, max\ data+1\right)\right)$
	i = (i + 1) mod (max data+1);
	j = (j + S [i]) mod (max data+1);
	Swap (S [i], S [j]);
	output1=S[S[i] + S [j]] mod (max data+1)]]
	shift S to the right or the left by k elements
	Swap (S [K], S [j]);
	output2=S[S[K] + S [j]] mod (max data+1)]]
	}

3.2.3 Encryption Algorithm

The block diagram of video compression and encryption is shown in Figure 2 and the description of each step in the proposed method for encryption is as the following:

- At the encoder side, the encryption algorithm as shown in Figure 2 started by reading video file and converts it into frames. Each frame in YUV color representation.
- The user secret key of the length N bits and initial condition parameters of chaotic maps are the inputs to the encryption algorithm. The chaotic maps used as another secret key with RC4 PRNG to increase the randomness on generating the j value that uses as index location pointer in RC4 PRNG algorithm and also used to generate the shift value of the state array.
- The I frames of the video sequence is selected for the encryption, because B frames depend on I or P frames, and P frames depend on I frames.
- During the compression process, each of video frames is divided into non-overlapping blocks of size 8×8. Then, for each block the DCT transformation and quantization processes are applied. The result is quantized DCT coefficients blocks.
- The quantized DCT coefficients blocks are arranged in a 1-D zig-zag sequence to perform the encoding and encryption process. The quantized DCT 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 are referred to AC coefficients. AC coefficients represent low and high frequency color change across the block. The DCT coefficients are selected to be encrypted by using the secret key and proposed method.
- For each I frames, extract the maximum value of DCT coefficients in the frame and it used with proposed method to encrypt DCT coefficients. The maximum value of DCT coefficients is considered as another key of proposed method and it send to the receiver with the key in secure channel.
- To increase security, the rotation technique used to rotor the Y blocks in I frame. The random direction is generated based on the Logistic chaotic map with initial condition.
- After encryption is done. The encrypted frame entered to the Entropy coding stage that it used to encode video frames. The output is encrypted MPEG bitstream.

3.2.4 Decryption Algorithm

The block diagram of video decoding and decryption is shown in Figure 3 and the description of each step in the proposed method for encryption is as the following:

- At the decoder side, Decryption algorithm as shown in Figure 3 started by loading the encrypted MPEG bitstreams file for MPEG decoding and decryption process.
- The same user secret key of the length N bits and initial condition parameters of chaotic maps are used for decryption algorithm.
- After the entropy decoding stage, all the blocks of size 8×8 in I frame are decoded. Then the inverse DCT transformation and inverse quantization processes are applied to each block. The result is blocks of IDCT coefficients.
- The IDCT coefficients will be decrypted by using proposed method and inverse rotation of Y blocks in I frame by using the same chaotic map. The result is the decrypted video frame. The output of the decoder is an original video.

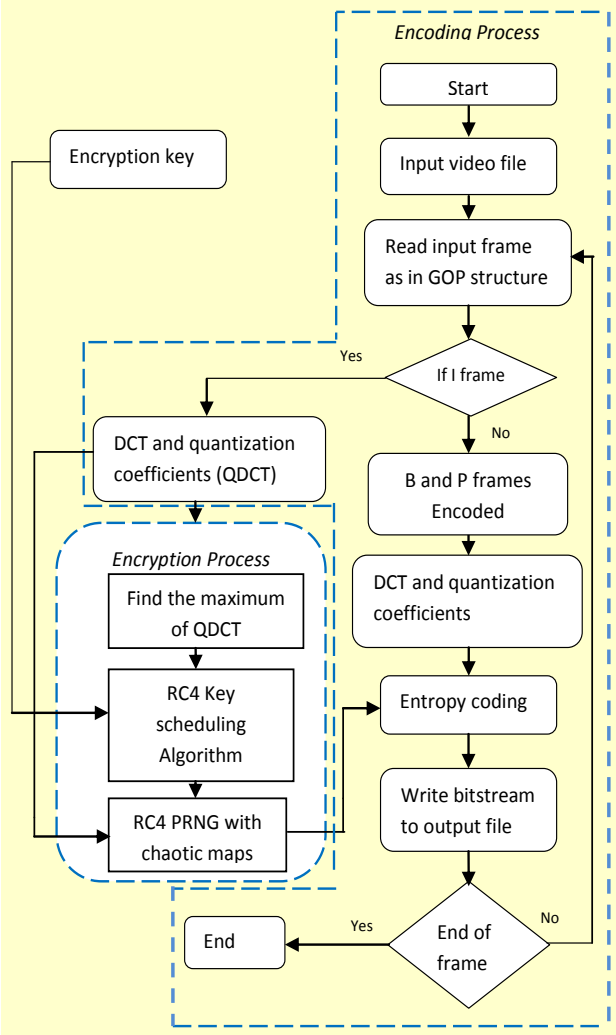


Figure 2 The block diagram of video compression and encryption

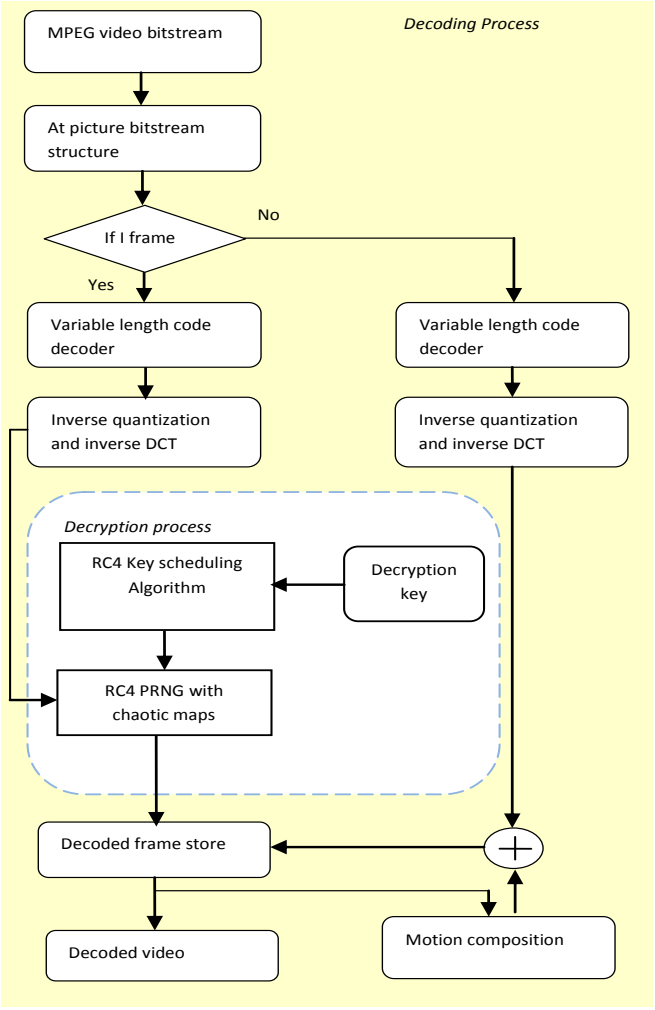


Figure 3 The block diagram of video decoding and decryption

4.0 RESULT AND DISCUSSION

To evaluate the performance in terms of the security, compression ratio, speed analysis of the proposed method series of experiments and analysis was implemented. In these experiments, four different video sequences, the Suzie, the Miss America, the Foreman and News sequences are used. The resolutions of QCIF video sequences are obtained in uncompressed 4:2:0 YUV format with frame size 144 ×176 pixels and frame rate is 30 frames per second. The first frame of the test video Suzie is shown in Figure 4a. a brief description for each video in these experiments as the following: the Susie sequence contains a large fast motion area, the miss America is a video sequence which contains several slow motion objects, the Foreman video sequence is talking head sequence with high motion foreground and a camera pan at the end and the News video sequence represent low motion objects. These videos are encoded using MPEG-2 encoder and encrypted/decrypted using the proposed method as shown in Figure 4b to 4d, with a key

=123456789ABCDEF4(in hexadecimal), x_0=0.1234, y_0=0.2134.

DCT coefficients are one of the important data in MPEG compressing process, the average color of each block in the frame represented by DC coefficient. DC coefficients also carry important visible information of video frames. The AC coefficients contain detailed frequency information of each block in video frame and motion vector difference contains dynamic information of frame.

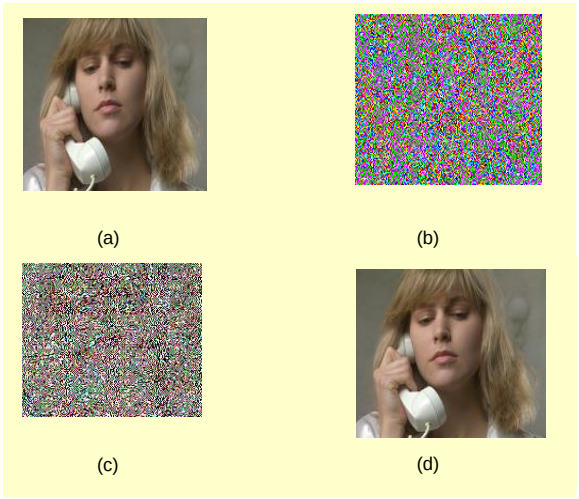


Figure 4 (a) Test video frame (suzie.yuv) (b) Encrypted frame by RC4 algorithm (c) Encrypted frame by proposed method and (d) Decrypted frame

The proposed encryption method encrypts all DCT coefficients in I frames to makes the frame unintelligible. To examine the performance of proposed method and compared with RC4 algorithm some experiment has been carried out in the terms of the encryption and decryption execution time, the size of the MPEG file before and after encryption, Compression Ratio, Peak-Signal-to-Noise-Ration (PSNR) and Structural Similarity Index Measurement (SSIM).

4.1 Compression Ratio Test

Data compression ratio is defined as the ratio between the uncompressed size and compressed size [2] as in (4):

$$CR = \frac{Uncompressed\ size}{Compressed\ size} \quad (4)$$

The encryption algorithms that used to encrypt multimedia data should not change the compression ratio or should at least keep the changes in a small range. However, encrypt all DCT will increase the burden in the encoding process, so there have an impact on the compression ratio. From Table 3, we can see the size of tested video was increased when all DCT coefficients in I frames are encrypted by using RC4 algorithm and proposed algorithm. Results analysis shows that the RC4 algorithm and proposed algorithm has an impact on the compression ratio because all the value of quantized DCT coefficients will be changed to large value after encrypted, so the bit stream will be also changed after entropy coding. From the results, it can be seen that the proposed algorithm has lower effects on the compression ratio than the original RC4 algorithm.

Table 3 Compression Ratio of encoded and encryption algorithms.

	Video sequence	Susie	Miss_Am	Foreman	News
	Original size	5.43	5.43	10.8	10.8
Encoding	Compressed size (MB)	0.72	0.57	2.08	1.41
	Compression ratio	7.54	9.52	5.19	7.65
RC4 algorithm	Size after encryption (MB)	2.84	2.72	6.23	5.57
	Compression ratio after encryption	1.91	1.99	1.73	1.93
Proposed Method	Size after encryption (MB)	2.26	2.29	5.33	5.16
	Compression ratio after encryption	2.40	2.37	2.02	2.09

4.2 Encryption Speed Test Analysis

To design a good video encryption algorithm, the encryption and decryption time is one of the important parameters that used to measure the performance of any encryption algorithm. Time taken by the proposed modified RC4 and RC4 algorithm to encrypt/decrypt various different sized videos such as Suzie, Miss_am, Foreman and News has been measured on an Intel Core i5 3 GHz CPU with 4 GB RAM running on Windows 7 Home Premium and using the MATLAB. The results are summarized in Tables 4 and 5 which clearly predicts an average encryption rate of the proposed method in complete encryption is 34 KB/second and it is better than encryption rate of the RC4 algorithm. The encryption time of the proposed algorithm is less than the encryption time of RC4 algorithm due to the round in the proposed algorithm depends on maximum value of plaintext.

Table 4 Total execution time of encoded and encryption algorithms

Video sequences	Encoding		RC4		MRC4		Decreasing time rate %
	Encoding time(s)	Encoding rate (KB/s)	Encoding + Encryption time (s)	Encoding rate (KB/s)	Encoding+ Encryption time(s)	Encoding rate (KB/s)	
Suzie.yuv (5.43 MB ,150 frame)	88.88	61.09	171.42	31.67	146.56	37.04	14.50
Miss_am.yuv (5.43 MB, 150 frame)	79.79	68.05	165.76	32.21	147.94	36.70	10.75
Foreman.yuv (10.8MB,300 frame)	216.31	49.90	384.97	28.05	358.41	30.13	6.89
News.yuv (10.8MB, 300 frame)	171.74	62.88	344.66	31.33	326.97	32.86	5.13

Table 5.The time taken to encode and encrypt I frame

Video sequences	Encoding time(s)	Encoding + Encryption (RC4) time (s)	Encoding + Encryption (MRC4) time (s)
Suzie.yuv	0.62	4.19	3.30
Miss_am.yuv	0.54	4.36	3.26

4.3 Peak-Signal-To-Noise-Ratio (PSNR)

PSNR is most commonly for video quality evaluation. PSNR measures the changes between each pixel in the original frames and its corresponding reconstructed frames without decryption of the video sequence. PSNR is easily defined by the Mean Squared Error (MSE) of two frames as in (5):

$$PSNR = 10 \times log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (5)$$

$$MSE = \frac{1}{N}\sum_{n}^{N-1}\left(x_f(n) - \hat{x_f}(n)\right)^2 \quad (6)$$

where MAX is the maximum possible pixel value of the image. The PSNR value approaches infinity when MSE is close to zero. Hence a higher PSNR value gives a higher video quality. Whereas, a small PSNR value high numerical differences between frames and hence low video quality. Table 6 shows the PSNR between the original I frame (Suzie.yuv) and encrypted frame when DCT components in I frame encrypted by using proposed method. From Table 6, we can see the average value of PSNR for all the sequences around 45.48 dB and the average PSNR of encrypted frames using proposed method and RC4 algorithm is 10.68 dB and 6.35 dB respectively. In average, the proposed modified RC4 algorithm and RC4 algorithm have low PSNR value and the security of digital video content is thus protected.

Table 6 PSNR for video sequences (Suzie.yuv) with and without encryption

Video sequences	PSNR of I frame (dB)								
	Without encryption			Encoding + Encryption (RC4)			Encoding + Encryption (MRC4)		
	Y	U	V	Y	U	V	Y	U	V
Suzie.yuv	42.54	46.83	47.07	6.12	6.48	6.44	6.81	14.85	10.84
average	45.48			6.35			10.68		

4.4 Structural Similarity Index Measurement (SSIM)

The SSIM was developed by Wang [14] and it provides a quality index measure of the similarity between two images. PSNR and MSE do not accurately depict human perception, SSIM was designed to represent the human perception and improve on methods like the PSNR.

The SSIM metric is calculated between two images x and y of common size N×N using equation (7)

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (7)$$

where μx and μy indicate the mean of x and y respectively, σx and σy indicate the standard deviation of x and y, σxy is the cross-correlation (inner product) of the mean shifted images μx and μy, and the C1 and C2 are positive constants. This formula applies to the luminance frames to evaluate the image quality with a maximum value of 1 showing excellent quality. The SSIM values of luminance of video sequences with and without encryption are shown in Table 7. From the results in Table 7, we can see the proposed algorithm has distorted the structural information present in the original video and the average SSIM value of video sequences without encryption is 0.9815, while it is 0.0046 and 0.0139 for RC4 algorithm and proposed algorithm respectively.

Table 7 Average SSIM: complete I frame encryption

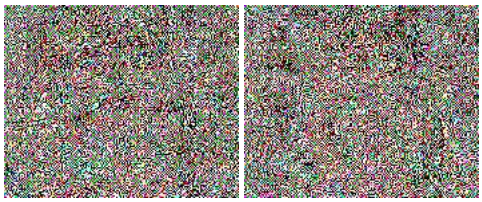
Video sequences	SSIM of I frame		
	Without encryption	Encoding + Encryption (RC4)	Encoding + Encryption (MRC4)
Suzie.yuv	0.9815	0.0046	0.0139

4.5 Key Sensitivity and Key Space Analysis

The total different number of keys which used in the encryption and decryption process is called key space. The key space of the encryption algorithm should be large enough to make brute-force attack ineffective [15]-[16]. The secret keys of the proposed algorithm are the two initial conditions of chaotic maps x0 and y0 both are real numbers, secret key of the length of 40 – 128 bits in RC4 algorithm and the maximum value of DCT coefficients. According to standard 64-bit IEEE floating point, computation precision of the floating point is 10^{15}, so number of possible values of *x0* is 1015 as well as y0. Thus, key space is $10^{15} \times 10^{15} = 10^{30}$. If we consider the initial key length is 128 bits and 8 bits of maximum value of DCT coefficients then we do have 2^{136} number of possible keys. Therefore, the key space of proposed algorithm is large enough so that the algorithm can be resistant to brute-force attack.

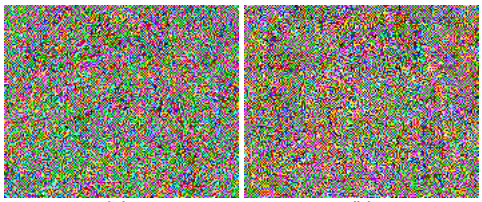
The secret keys in the encryption algorithm should be very sensitive to any change of the key to make brute force attacks is not possible. A test video clip called "Suzie.yuv" is used to test the sensitivity to secret keys. The test video was compressed and encrypted using the MPEG-2 encoder. The secret key {123456789ABCDEF4}, x_0=0.123, y_0=0.213 was chosen to encrypt test video. The bitstream decrypted by using the correct key produces the original frame as shown in Figure 5. The following experiments were conducted to analysis the encryption effect. In first test, the encrypted bitstream was decoded normally at the decoder side without applying decryption algorithm. This test demonstrates the effect of decoding an encrypted bitstream using the MPEG-2 decoding process. In second test, the encrypted bitstream is decrypted using a randomly generated key (17460FD0). This test presents the case where an adversary attempts an attack by decrypting the bitstream using a randomly key. In third test, the encrypted bitstream is decrypted using the key {123456789ABCDEF5}, which is different from the correct key by one digit (for example, the last digit being 4 instead of 5). This is to show the scenario where an attacker was able to find out most parts of the correct key value. It is clear from the Figure 6 to Figure 8, all tests produced totally scrambled images and video content do not have any meaning. This indicate that the proposed method has high sensitivity to any changes occurs to the secret key.

Figure 5 Sample frames from video Suzie sequences. (a) Frame 1. (b) Frame 50



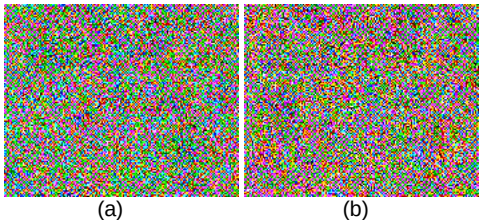
(a) (b)

Figure 6 Sample frames from video Suzie sequences decoded without decryption by the MPEG 2 decoding process. (a) Frame 1. (b) Frame 50



(a) (b)

Figure 7 Sample frames from video Suzie sequences obtained through decryption using key (17460FD0). (a) Frame 1. (b) Frame 50



(a) (b)

Figure 8 Sample frames from video Suzie sequences obtained through decryption using key {123456789ABCDEFG.x_0=0.1234, y0=0.2134}. (a) Frame 1. (b) Frame 50

5.0 CONCLUSION

In this paper, the modification of RC4 algorithm has been proposed to suppress bitstream size increment in video encryption which utilizes circular shift and chaotic maps to make the stats matrix in RC4 algorithm depending on maximum plaintext. From the results, it can be seen that the proposed method provides good encryption quality. The time taken for video encryption using proposed method is relatively less in comparison with the RC4 algorithm. The proposed method has large key space and they sensitive to any changes occur in secret key, which makes the proposed algorithm robust against any attack. The proposed method has lower effects on the compression ratio than the original RC4 algorithm. Results verify the proposed scheme has distorted the structural information present in the original video. Therefore, the proposed method has a high security which is due to characteristics of the chaotic system and it is suitable for secure video storing and transmission.

Acknowledgement

This study was supported by the Universiti Kebangsaan Malaysia (UKM) through research grants LRGS/TD/2011/UKM/ICT/02/02.

References

[1] Shin, S. U., Sim, K. S., and Rhee, K. H. 1999. A Secrecy Scheme for MPEG Video Data Using the Joint of Compression and Encryption. *In: Proc. Information Security, Lecture Notes in Computer Science*. 1729: 191-201. Berlin Springer-Verlag.

[2] Lian, S. 2009. *Multimedia Content Encryption: Techniques and Applications*. Boca Raton, FL, USA: CRC Press. 89-107.

[3] Shi, C., and Bhargava, B. 1998. An Efficient MPEG Video Encryption Algorithm. Proc. *on Reliable Distributed Systems. Seventeenth IEEE Symposium*. West Lafayette, IN. 20-23 October 1998. 381-386.

[4] Tang, L. 1996. Methods for Encrypting and Decrypting MPEG Video Data Efficiently. *Proc. on MULTIMEDIA '96 Proceedings of the fourth ACM international conference on Multimedia*. 219-229.

[5] Zeng, W., and Lei, S. 2003. Efficient Frequency Domain Selective Scrambling of Digital Video. *Multimedia, IEEE Transactions*. 5(1): 118-129.

[6] Sikora, Thomas. 1997. MPEG Digital Video-coding Standards. *Signal Processing Magazine, IEEE*. 14(5): 82-100.

[7] Le Gall, D.1991. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*. 34(4): 46-58.

[8] Weerasinghe, T. D. B. 2012. Analysis of a Modified RC4 Algorithm. *International Journal of Computer Applications*. 51(22): 12-16.

[9] Mousa, A., and Hamad, A. 2006. Evaluation of the RC4 Algorithm for Data Encryption. *International Journal of Computer Science and Applications*. 3(2): 44-56.

[10] Pareek, N. K., Patidar, V., and Sud, K. K. 2006. Image Encryption Using Chaotic Logistic Map. *Image and Vision Computing*. 24(9): 926-934.

[11] Kocarev, L., and Jakimoski, G. 2001, Logistic Map as a Block Encryption Algorithm. *Physics Letters A*. 289(4): 199-206.

[12] Alvarez, G., and Li, S. 2006. Some Basic Cryptographic Requirements for Chaos-based Cryptosystems. *International Journal of Bifurcation and Chaos*. 16(8): 2129-2151.

[13] *Mao, Y., & Chen, G.* 2005. *Chaos-based Image Encryption.* In Handbook of Geometric Computing, Springer Berlin Heidelberg. 231-265.

[14] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *Image Processing, IEEE Transactions*. 13(4): 600-612.

[15] Chen, G., Mao, Y., & Chui, C. K. 2004. A Symmetric Image Encryption Scheme Based on 3D Chaotic Cat Maps. *Chaos, Solitons & Fractals.* 21(3): 749-761.

[16] Munir, R. 2012. Security Analysis of Selective Image Encryption Algorithm Based on Chaos and CBC-Like Mode. *Telecommunication Systems, Services, and Applications (TSSA), IEEE International Conference*. Bali. 30-31 October 2012. 142-146.