PERMUTATION INDUCED SHUFFLING BASED BLOCK CIPHER FOR REAL-TIME APPLICATIONS

***N. Radha*[1], *M. Venkatesulu*[2]**

[1, 2] Department of Computer Applications, Kalasalingam University, Krishnankoil, Srivilliputtur (via), TamilNadu-626190. India

[1]radha_athithan@yahoo.com, [2]venkatesulum2000@gmail.com

ABSTRACT

We proposed a novel non-conventional full encryption approach designed to accommodate fast performance and security with low computational complexity for real-time video communications. The proposed cipher employs the confusion and diffusion component simultaneously through the shuffler operator S and confusion component through the randomized substitution R in a binary tree. A pseudo random integer sequence π generates the round keys and also used to define the non-linear confusion and diffusion components. The advantages of these components are 1) they do not require storage space, when compared to generation and storage of S-boxes in other systems. 2) they are faster. We analyzed our approach in terms of efficiency and security from a cryptographic view point. We analyzed performance by comparing with the symmetric algorithms AES and RC6. Experimental results showed that the proposed algorithm works faster in terms of encryption time and frame rate and thus suitable for rendering the data in real-time by the receiver's playback. Histogram analysis, correlation analysis and differential cryptanalysis proved the efficiency of the cipher.

Keywords: Block cipher, Naive approach, Substitution, Diffusion, Shuffling, Real-time.

1.0 INTRODUCTION

The widespread usage of multimedia data in the internet makes media content protection important. Internet telephony, internet conferencing, internet security monitoring, video teleconference, pay-per-view (PPV), video-on-demand (VOD), medical image transmission, interactive video games are some of the real time applications that require confidentiality. To accommodate the confidentiality in various applications, multiple levels of security are desirable. Mathematical algorithms called cryptographic primitives found the basis of the security [1]. Cryptographic primitives are cryptographic algorithms such as encryption schemes, hash functions and digital signature schemes. The increased popularity of multimedia applications places a great demand on the security of multimedia content during transmission. Thus real-time applications brought new challenges to the protection of multimedia content confidentiality. To protect the user's sensitive data in real-time, a number of techniques have been proposed in the literature which can be classified into three categories: i) Full encryption approach ii) Selective encryption approach and iii) Combined compression-encryption approach.

Full encryption approach is a way to accomplish content confidentiality by encrypting the raw data directly or compressed data directly using traditional ciphers such as DES, 3DES, AES, IDEA or RSA which all incur high processing and computational complexity. This is the most secure approach for the encryption of data, encrypting the whole stream, and it is not format compliant. Selective encryption approach encrypts only parts of a compressed multimedia content to reduce the computational requirements of client devices in real-time applications. The major issue is to select the important parts that will be encrypted i.e) the parts whose encryption will guarantee that the adversary cannot recover useful information about the original multimedia data. These algorithms have low computational complexity and security and the advantage is greater speed and format compliant. Combined compression-encryption approach combines the compression process and encryption process into a single step. This is achieved by entropy coders which use secret keys to encode the data using multiple Huffman Table (MHT), randomized arithmetic coding (RAC), Arithmetic coding with key-based interval splitting approach (KSAC).

General-purpose encryption standards such as RC4 and AES for real time video streaming is studied in [2]. It is a good idea to apply a conventional encryption algorithm for video transmission which ensures confidentiality and security at a high level [3]. As the full encryption approach provides very high security compared to partial/selective encryptions, we propose a novel non-conventional full encryption approach designed to accommodate fast performance and security with low computational complexity which brings acceptable interval between encryption and decryption to suit the necessity of real-time applications. We describe the cipher which employs the confusion and diffusion component simultaneously through the shuffler operator S and confusion component through the randomized substitution R in a binary tree. We measure the performance in terms of encryption time and frame rate and thus suitable for rendering the data in real-time for video transmissions. Histogram analysis, correlation analysis and differential cryptanalysis prove that the cipher is secured.

The rest of the paper is organized as follows: Section 2 describes the related study for the proposed work. Section 3 describes the proposed algorithm. Section 4 deals with the mode of implementation. Section 5 is devoted to experimental results. Section 6 deals with the security analysis. Section 7 gives the conclusion.

2.0 RELATED WORK

2.1 Block Encryption

Symmetric encryption algorithms are a class of algorithms that use the same cryptographic keys for both encryption and decryption. The two main types of modern symmetric encryption algorithms are Block ciphers and Stream ciphers. The objective of block cipher is to provide confidentiality or secrecy to the data in communication transactions. Block ciphers plays larger role in the internet, wireless networks and computing devices against active or passive attacks. A block cipher is a method of encrypting a block of plaintext using a cryptographic key and an algorithm. It is designed by two principles: diffusion and confusion. Confusion refers to making the relationship between ciphertext and key as complex as possible and diffusion refers to making the relationship between plaintext and ciphertext as complex as possible. Block ciphers are constructed using Fiestel and SPN (Substitution-Permutation Network) structures. In fiestel structure, the encryption process and decryption process are similar; they require only a reversal of the key schedule. SPN uses S-boxes and P-boxes that transform blocks of plaintext to ciphertext. Examples of Fiestel structures are Blowfish, DES, CAST-128, Camellia, KASUMI, FEAL, XTEA, Twofish, MARS, Lucifer, TEA, RC5, GOST 28147-89 and Triple DES. Examples of SPN structures are SAFER, AES, SHARK, Square, 3-way and others.

Modern Symmetric block ciphers are based on the substitution-permutation network introduced by Claude Shanon [4] in 1949. According to Shanon, the ciphertext completely obscure the statistical properties of the original message through confusion and diffusion. Confusion can be accomplished by using a complex substitution algorithm and diffusion by permutation algorithm. Substitution algorithm generally replaces certain bits with other bits and permutation algorithm manipulates the order of bits which provides shuffling of bits. A well-designed SPN block cipher has several alternating rounds of S and P boxes which satisfies Shannon's confusion and diffusion properties. Many block ciphers based on key-dependant S-boxes and P-boxes have been investigated for years in order to protect communication security [5-10]. One of the desirable properties of the typical block cipher is the avalanche criterion, an effect in which a small change in the key or plaintext causes changes in the ciphertext significantly. If the cipher does not exhibit the avalanche effect to a certain degree, the adversary make predictions about the input and the key and he partially or completely breaks the cipher.

2. 2 Video Encryption Approaches

Many video encryption schemes have been studied in the literature and they optimize the encryption and display process with respect to the encryption speed. *Naïve algorithm guarantees* the security to the whole stream by standard encryption algorithms such as AES [11] or Triple DES [12]. However, Triple DES is not applicable for real time encryption because of the delay introduced in the encryption process. *Pure Permutation* [13] confuses the bytes within a frame and is susceptible to known-plain text attack. In [14], the *Zigzag permutation* maps the 8X8 block to a 1X64 vector utilizing a random permutation. In [15-17], the VEA was proposed which encrypts the sign bit of DCT coefficients and motion vectors using traditional symmetric key cryptography. Several selective encryption methods encrypt at different levels have been proposed in [18-20]. Some of the proposed video encryption based on chaos is studied by [21-25]. These algorithms are applied to video data.

The algorithm in [26] shuffles the video frames along with the audio, and then AES is used to selectively encrypt the sensitive video code words. A light weight cipher which provides media QoS and security was studied in [27]. In [28], fast and secure real-time video encryption was proposed which exploits DCT coefficients and achieves computational efficiency. A Dynamic Encryption Algorithm for the Real-Time Applications (DEA-RTA) was studied in [29]. To reduce the computational overhead yet achieving security for more complex data, modification with the ShiftRow module in AES is studied in [30].

Li et al [25] studied CVES, a chaotic video encryption scheme, encrypts any format of video files and the encrypted file cannot be seen without decryption. This implies that it is impractical to perform rate alteration exactly on the encrypted data and the processing would require the keys to decrypt the data and get back the original content. Our proposed scheme uses the idea similar to this. To view the unencrypted data, a special decoder is essential. FEA-M, a fast encryption for multimedia was proposed in [31]. A block cipher of 512 bits suitable for real-time multimedia is studied in [32].

3.0 DESCRIPTION OF THE PROPOSED ALGORITHM

3.1 Encryption Process Model

Fig. 1. shows the encryption overview of the proposed algorithm.

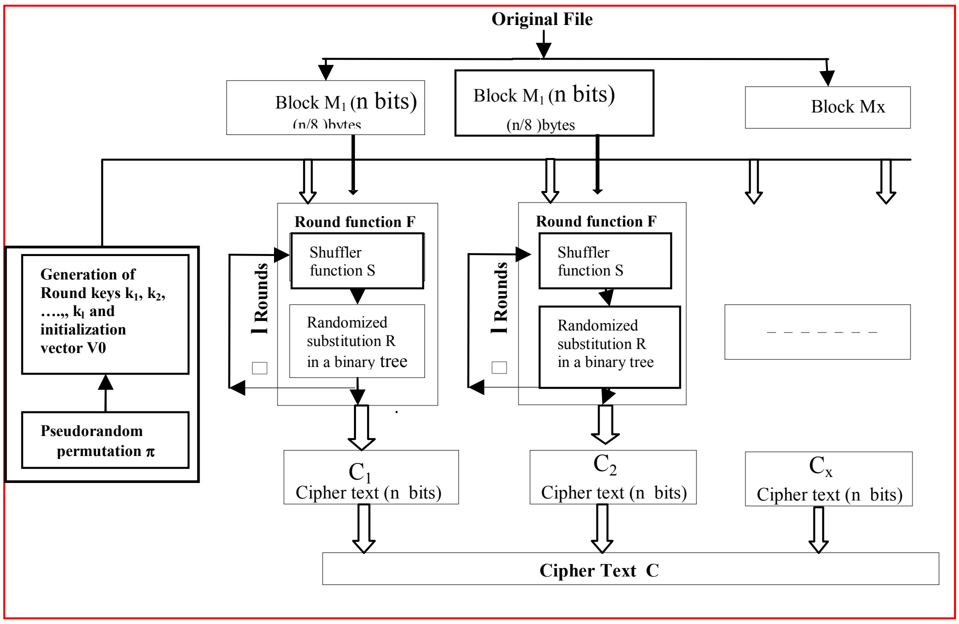


Fig. 1. Encryption Process overview

3.2 Key hierarchy and key generation

The key hierarchy is shown in Fig. 2.

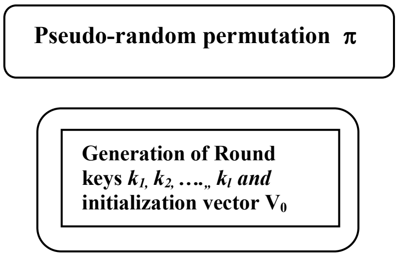

Pseudo-random permutation π
Generation of Round keys $k_1, k_2, \ldots,, k_l$ and initialization vector V_0

Fig. 2. Key hierarchy

3.2.1 Choose a pseudo random permutation of integer sequence, π

Choose a pseudo random permutation π of the position set *{1, 2, ..., n}* say, $\pi = \{ p_1, p_2, p_3, \ldots p_n\}$ Є *{1,2, ...n} of block size n.* Each p_i is represented by 2^l bits where *l=3, 4,…….n.*

3.2.2 Generation of Round keys from the integer sequence π

Generate the round keys $k_1, k_2, \ldots,, k_l$, each of size *n*, from the integer sequence π. Represent each integer in π as a binary number and collect first LSB from each number to form key k_1, and collect second LSB from each number to form key k_2 and so on. For example, for the block size n = 8, there are 3 round keys k_1, k_2 and k_3. For π= {6, 5, 7, 4, 2, 3, 1, 0}, it is represented in binary as {110, 101, 111, 100, 010, 011, 001, 000} and choose k_1 = 01100110, k_2 = 10101100 and k_3 = 01100110. In addition to the round keys, the sender generates the initialization vector V_0 randomly from the round keys.

3.3 Generation of shuffler operator S (confusion and diffusion component) from π

A shuffler operator S, based on the distance between the permuted positions is defined which provides substitution, then XORing with k_i dynamically on the plaintext block *M* yields a pseudo-ciphertext *C'*.

The sender generates the shuffler operator S from π and applies the operator on the plaintext M and computes S(M) by the following pseudocode:

```
        Let $M = (m_1, m_2, m_3, \ldots, m_n)$ and $S(M) = (b_1, b_2 \ldots b_{n-1}, b_n)$
        Let $\pi = \{ p_1, p_2, p_3, \ldots p_n\}$
  S on M is represented as:
(1)
    for $0 < i < n$,
       Let $j = p_i$
       Let $\pi(i) = j$,
                 if $|i-j|$ = even then
                      $b_j = m_i$,
                 else
                      $b_j = 1 - m_i$
                 endif
    endfor
```

where $|i-j|$ is the distance between two integers *i* and *j*.

It is represented pictorially in Fig. 3.

Let π = {4, 1, 8, 3, 2, 7, 5, 6} and M = {0, 1, 1, 0, 0, 1, 1, 1}
Then, for $\pi(i) = j$, we have $\pi(1) = 4$, $\pi(2) = 1$, $\pi(3) = 8$, $\pi(4) = 3$, $\pi(5) = 2$, $\pi(6) = 7$,
$\pi(7) = 5$, $\pi(8) = 6$
Then, b_4= {1} |1-4| = 3 = odd
Then, b_1 = {0} |2-1| = 1 = odd
Then, b_8 = {0} |3-8| = 5 = odd
Then, b_3= {1} |4-3| = 1 = odd
Then, b_2= {1} |5-2| = 3 = odd
Then, b_7 = {0} |6-7| = 1 = odd
Then, b_5= {1} |7-5| = 2 = even
Then, b_6= {1} |8-6| = 2 = even
Thus *S (M) = (0, 1, 1, 1, 1, 1, 0, 0).*

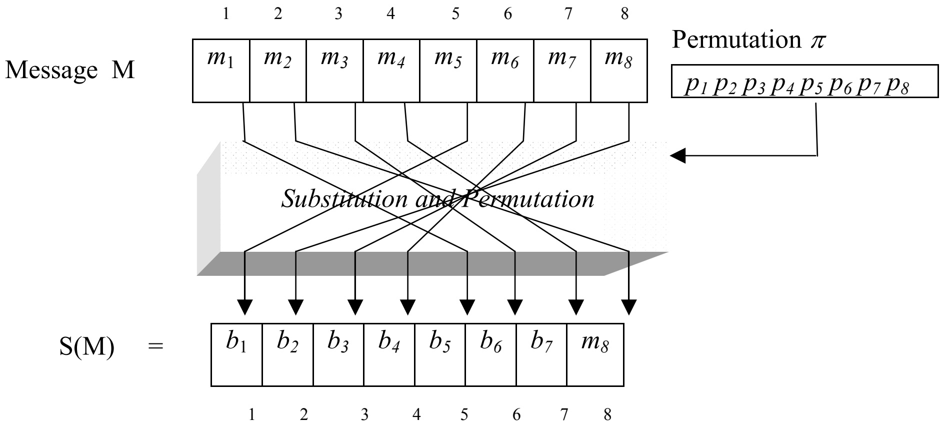

1 2 3 4 5 6 7 8
Message M
m_1 m_2 m_3 m_4 m_5 m_6 m_7 m_8
Permutation π
$p_1\,p_2\,p_3\,p_4\,p_5\,p_6\,p_7\,p_8$
Substitution and Permutation
S(M) =
b_1 b_2 b_3 b_4 b_5 b_6 b_7 m_8
1 2 3 4 5 6 7 8

Fig. 3. Shuffler operator *S* on plaintext *M*

In fact, according to Substitution-Permutation Network (SPN), the shuffler operator S defines two components, substitution and permutation dynamically, which performs an important security role in the algorithm.

The advantages of these components are 1) They do not require much storage space, when compared to generation and storage of S-boxes in other systems. ii) They are faster. iii) For any pseudo random permutation π, the shuffling operator S on message M is non-linear.

The shuffler operator *S* defines a non-linear substitution component, if the permutation on $\pi(i)=j$ such that $|i-j|$ is odd for at least one *i.* If $|i-j|$ is odd, perfect non-linear functions exist.

3.4 Randomized substitution R in a binary tree (confusion component) using π

Arrange the bits from the previous stage in the given order into a complete binary tree starting with the left most bit as the root node, the next two elements form the left and right children of the root and so on, always going from higher level to the next lower level and from left to right of the tree. Every node undergoes substitution such that strict avalanche criterion is achieved with the following pseudo code:

```
Let M = $(m_1, m_2, m_3, \ldots, m_n)$ and $R(M) = (b_1, b_2 \ldots b_{n-1}, b_n)$
Let $\pi = \{ p_1, p_2, p_3, \ldots p_n\}$
 R on M is defined as follows:                                        (2)
      Let i = 1, 2, 3 …. n
       Let j = $\{ p_1, p_2, p_3, \ldots p_n\}$
      Let $\pi(j) = i$,
      for $1 < j < n$,
            for $1 < i < n$,
                  Define $b_i$ = ( the length of the path between $i$ and $j$ ** ) % 2 $\oplus$ the value
                              of $m_i$'s along the path from node $j$ to node $i$
                  Set $m_i = b_i$
            endfor
      endfor
```

** The path between *j* and *i* is traversed as follows:
Case (1) : If the node '*i*' is a descendant of the node '*j*', then node values of the sub tree rooted at *j* are traversed using pre-order traversal.
Case (2) : If the node '*i*' is not a descendant of the node '*j*', then we traverse from *j* to *i*, covering all the nodes along the path, starting from j, then the nearest node, the second nearest node and so on (using in-order or post-order traversals)

3.5 Encryption and Decryption

3.5.1 Split into Blocks

The input file is split into blocks, each block is assigned $n=2^l$ bits where l=3, 4,……. Let the blocks be M_1, M_2, M_3, M_4, …, M_x. Each block has n/8 bytes of plaintext P_1, P_2,…….$P_{n/8}$, which forms a binary message block M_j of size n. Let the bits in the block M_j be m_1, m_2, m_3, …., m_n. Let the block size may be 16, 32, 64, 128, 256, 512,….. bits.

3.5.2 Round function

For each block, get the Pseudo-ciphertext *C''* as follows,

$$C' = S\,(M \oplus k_i) \oplus k_i \quad where\ 1<i<l$$
$$C'' = R\,(C')$$

(3)

The round function F for a block is shown in fig. 4.

3.5.3 Repeat the step 3.4.2 l rounds

The Encryption process performs *l* rounds of same transformations applied to the Pseudo-ciphertext repeatedly and finally gets the ciphertext *C* of size *n*. The encryption and decryption algorithm is shown in fig. 5.

4.0 MODE OF IMPLEMENTATION

The proposed algorithm is implemented in CBC mode. We propose a naive encryption scheme to encrypt a compressed bit stream independent of compression algorithms, which treats the input as a traditional data stream such as text and encrypts the entire data stream. Our encryption and decryption algorithm design should satisfy the similarity of encryption (E) and decryption (D) for ease of implementation. Suppose M_1, M_2, … M_x have been transformed to $C_1, C_2, \ldots C_x$ using k_i keys for l rounds according to formula (3). The same transformations with R^{-1} and S^{-1} are performed on $C_1, C_2, \ldots C_x$ as shown in fig. 5 to get the plaintext $M_1, M_2, \ldots M_x$.

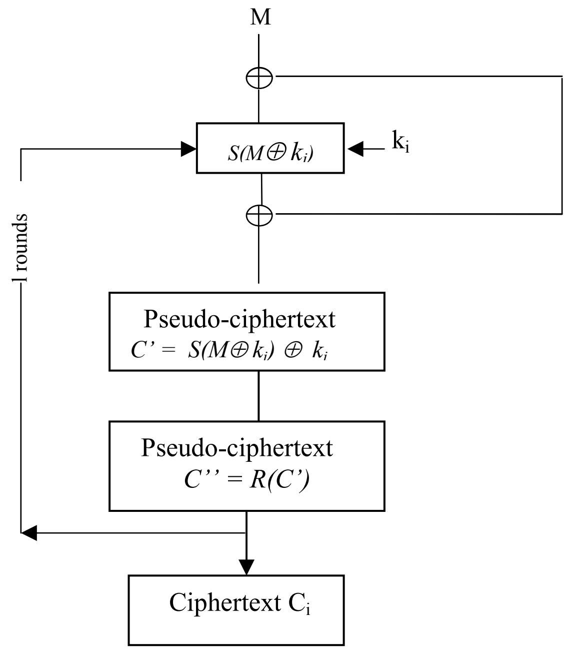


Fig. 4. Round function F for a block

4.1 Similarity of encryption and decryption in CBC mode:

Each plaintext is encrypted into ciphertext in the following manner,

$C_1 = R(S(((M_1 \oplus V_0) \oplus k_i)) \oplus k_i))$ *where* $1<i<l$ *with* l *rounds*

(4)

$C_2 = R(S(((M_2 \oplus C_1) \oplus k_i)) \oplus k_i))$ *where* $1<i<l$ *with* l *rounds*

.

.

$C_x = R\ (S(((M_x \oplus C_{i-1}) \oplus k_i)) \oplus k_i))$ *where* $1<i<l$ *with* l *rounds*

Each corresponding ciphertext is decrypted into plaintext in the following manner,

$= R^{-1}(C_1)$

(5)

$= S(R^{-1}(C_1) \oplus k_i)$

$= S((R^{-1}(C_1 \oplus k_i) \oplus k_i) \oplus V_0)$

$= M_1$

$= R^{-1}(C_2)$

$= S(R^{-1}(C_2) \oplus k_i)$

$= S((R^{-1}(C_2 \oplus C_1) \oplus k_i) \oplus k_i)$

$= M_2$

.

.

$= R^{-1}(C_x)$

$= S(R^{-1}(C_x) \oplus k_i)$

$= S((R^{-1}(C_x \oplus C_{x-1}) \oplus k_i) \oplus k_i)$

$= M_x$

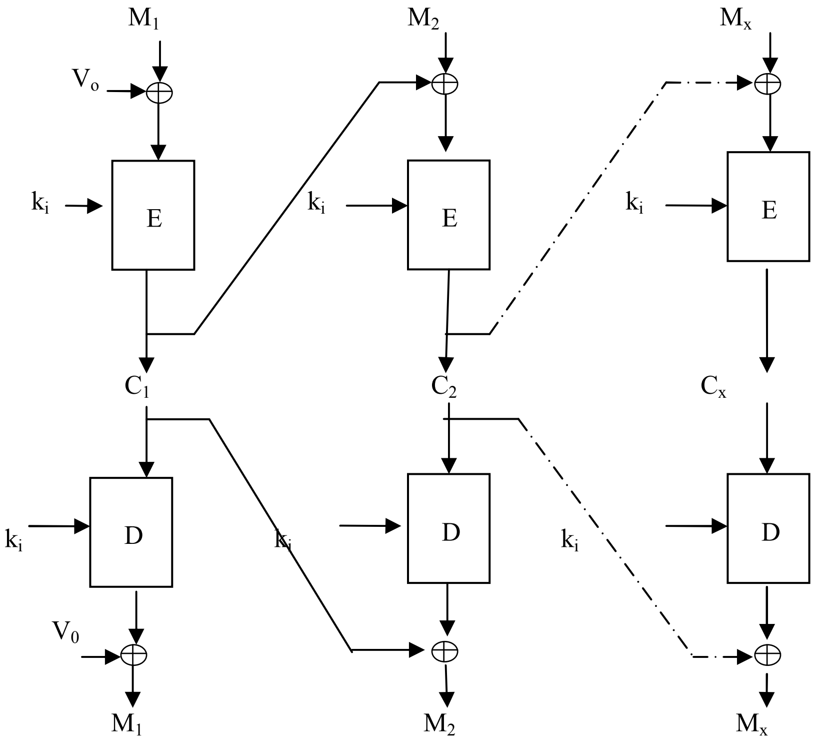


Fig. 5. Encryption and decryption process

4.2 Confusion and diffusion

The shuffling operator S induces the confusion and diffusion simultaneously into our algorithm design. The shuffling operator satisfies the almost nonlinear permutation *f:* GF(2^n) -> GF(2^n) such that, for any two pairs of specific inputs with the identical difference, the equivalent output pairs takes absolutely 2^{n-1} dissimilar non-zero values to protect it from differential cryptanalysis.

The random substitution R in a binary tree induces the confusion into our algorithm design. It is new data structures that traverses the binary tree in-order, pre-order or post-order for finding the length of the path for substitution and acquire good diffusion, since each plaintext or key bit modify half of the output bits.

5.0 EXPERIMENTAL RESULTS

We used our algorithm to encrypt and decrypt files such as document, image, audio and video. The experimental environment are as follows:
CPU: Intel Core 2 Duo E7200 @ 2.53GHz, 0.99 GB RAM; Operating system: Windows XP; Java and Netbeans IDE.

5. 1 Performance Analysis

In real-time applications, the transmission and delivery of data over a network requires that the data are sent in an admissible delay for the video frames to be displayed at a certain frame rate. The encryption and decryption algorithm for real time video streaming using AES, RC4 and XOR is studied in [31] [32] and the results showed that AES algorithm encrypts 29.3 frames per second (fps) for JPEG, while the original JPEG player encrypts 23.5 fps up to 28 fps[32]. The result also exhibited the delay produced from encryption and compression is nearly 0.031 second which is satisfactory for video transmissions [32]. Thus AES can be acceptable for real time video communication with little processing cost [31].

We measured the performance in terms of *encryption time and frame rate* for the video files. The performance of the proposed algorithm is analyzed with two encryption algorithm, AES and RC6. The various parameters for block ciphers are tabulated in table 1. The comparison has been conducted by simulating the encryption algorithms using their standard specifications with less optimization for the various *block sizes with* the *CBC mode* to process input video files of varying sizes and contents to evaluate the algorithms.

Table 1: Block cipher parameters

Parameters	AES	RC6	Proposed
Block size	128, 160, 192, 224, 256	64, 128, 256	8, 16, 32, 64, 128, 256, 512, ….
No of rounds	128:10 192:12 256:14	0, 1, 2, … 255	128 :8 256 :9 512:10 1024:11 2048:12
Key length	128, 192, 256	0, 1, 2, … 255	8, 16, 32, 64, 128, 256, 512, ….
Look-up tables	Large	-	-
Operations	Addroundkey(), sub-bytes(), shiftrows(), mixcolumns(), rotword(), ., ⊕, ⊗	+, -, ⊕, <<<, >>>, *	⊕, Array shuffling, tree traversal
Key expansion	More time	Less time	Less time

5. 1. 1 Performance with encryption time

The encryption time is the time taken to encrypt the entire file measured in milliseconds. Table 2 and Fig. 6 shows the results of the encryption time using AES, RC6 and the proposed algorithm for the input file sizes of 2, 5, 10, 18 and 25 MB for the block size of 128 bits. For the given input, AES undergoes 10 rounds, RC6 undergoes 20 rounds and the proposed algorithm undergoes 8 rounds with the key size of 128 bits. The key expansion for AES has key scheduling consists of rotate, Rcon and S-box operations which take more time in software implementation. The key expansion for RC6 has key scheduling consists of rotate operation which takes less time. The proposed algorithm has keys generated from the integer sequence which also takes less time. The proposed algorithm shows good performance compared to other algorithms.

Table 2: Encryption time using AES, RC6 and proposed algorithm for the block size of 128 bits.

Input file size(MB)	AES (ms)	RC6 (ms)	Proposed (ms)
2	1245	1789	758
5	1437	1974	920
10	1612	2184	1069
18	1684	2250	1143
25	1745	2313	1204

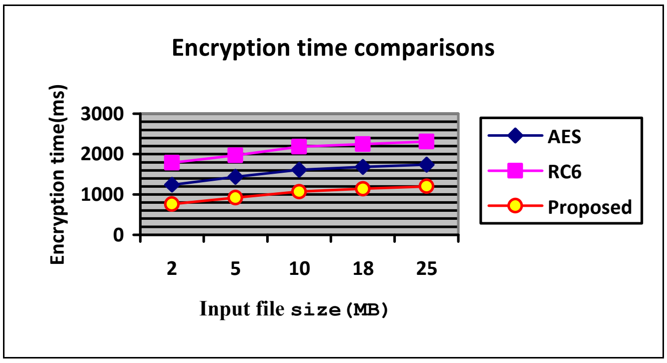


Fig. 6. Performance comparison of encryption time

5. 1. 2 Performance with Frame rate

The encryption speed measured in KB/s is the volume of data encrypted by the algorithm per second. Using the volume of data encrypted, frame rate is calculated for the algorithms. We have measured the encryption speed and the frame rate for the video file of size 25 MB for the block sizes 128, 224 and256 bits. Table 3 and Fig. 7. shows the frame rate of the algorithms. The frame rate increases for the higher block sizes. AES performs 30 frames/second, RC6 performs 28 frames/second and the proposed algorithm performs 36 frames/second for the key size of 128 bits for the block of 256 bits. Thus the proposed algorithm shows good performance compared to other algorithms.

From Figs. 6 and 7, it is noticed that

- The encryption speed and time are faster for the proposed algorithm because the operations used involve array implementations with low computational effort whereas AES and RC6 involve high computational effort.
- Because of the low computational effort, the proposed algorithm encrypts 36 frames/second, thus supporting sending and receiving of data stream at a bounded frame rate which is suitable for the receivers playback in real time encryption and transmission.
- Thus the proposed system is applicable to the scenarios of multimedia in real time such as pay-per-view channel protection, video-on-demand protection and video conferencing protection with high-level security.

Table 3: Frame rate comparisons using AES, RC6 and proposed algorithm for the file size 25 MB for the block sizes 128, 224 and256 bits.

Block size	AES (fps)	RC6 (fps)	Proposed (fps)
128	23	19	27
224	27	24	31
256	30	28	36

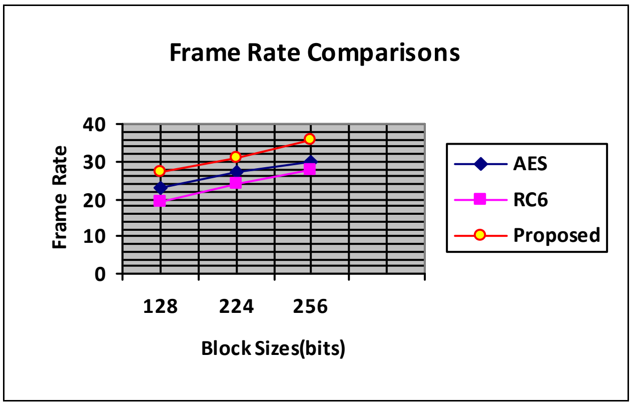


Fig. 7. Performance comparison with frame rate

5. 1. 3 Performance with the encrypted images and its histograms

Fig. 8. shows the encrypted images and its histograms for the AES, RC6 and the proposed algorithm for the 256 gray level lena image. The original image histogram contains sharp raise and declines as presented in fig. 8. B. Like AES and RC6, it is clear that the proposed encrypted image histogram in fig. 8. H. is fairly smooth and different from the respective histogram of the original images as presented in fig. 8. B and hence does not give any hint to apply any statistical attack on the proposed algorithm. The uniform distribution in cipher-image histogram is validated by the chi-square test using formula (6).

$$x^2 = \sum_{k=1}^{256} \frac{(v_k - 256)^2}{256} \tag{6}$$

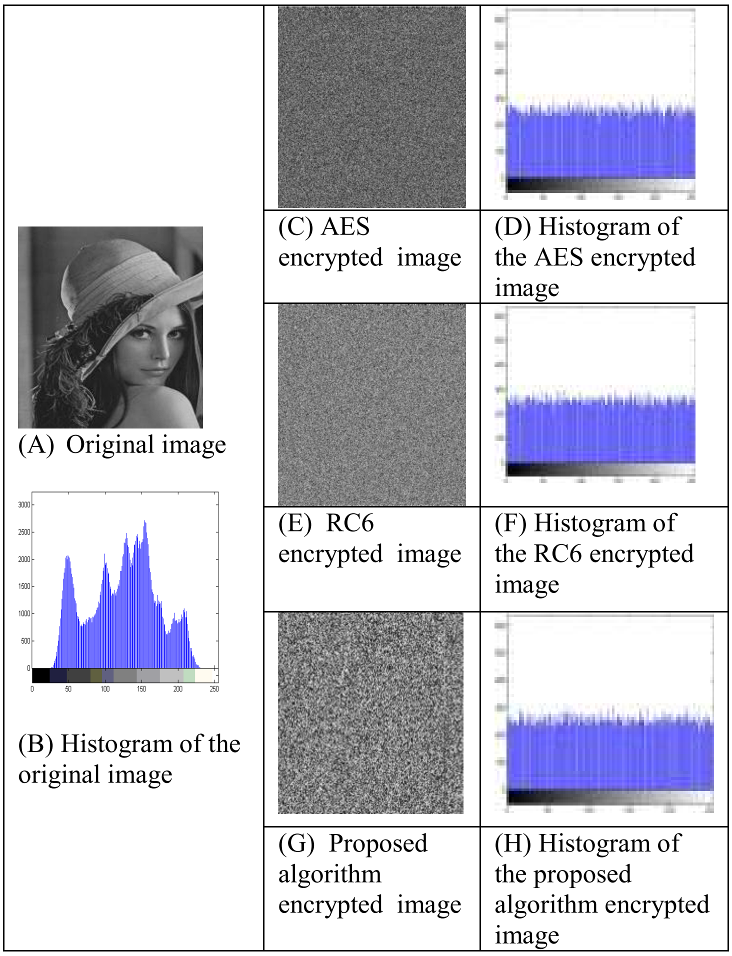
where k is the total gray levels, v_k is the examined frequency of each gray level in the range 0-255 and the expected state of each gray level is 256. Expecting a level of 0.01, the calculated value for χ^2 (255, 0. 01) =289, χ^2_{test} = 212 and found that $\chi^2_{\text{test}} < \chi^2$ (255, 0. 01), implies that the null hypothesis is not rejected and the proposed histogram has uniform distribution.

6.0 SECURITY ANALYSIS

The proposed algorithm satisfies the confusion and diffusion theory proposed by Claude Shanon [33]. The proposed cipher relies on the pseudorandomness of the integer sequence π, the non-linear function through shuffler operator S and diffusion through random substitution R in a binary tree, in addition to primitive XOR operation in each round. Some of the security features incorporated in the cipher is discussed below:

6.1 Brute force attack

The proposed cipher has n! permutations of keys. Thus it has distinct combination of keys when *n* is huge and makes brute force attack infeasible.



(A) Original image

(B) Histogram of the original image

(C) AES encrypted image

(D) Histogram of the AES encrypted image

(E) RC6 encrypted image

(F) Histogram of the RC6 encrypted image

(G) Proposed algorithm encrypted image

(H) Histogram of the proposed algorithm encrypted image

Fig. 8. Encrypted images and its histograms

6. 2 Theorem 1: (Non-Linearity of the shuffling operator S)

The perfect non-linear function exists if at least one of |i-j| is odd.

Proof:

For any Pseudo random permutation π, the shuffling operator *S* on message *P*, is non-linear. Let f be a function from GF(2^n) ----> GF(2^n), f is called non-linear if

$$S\,(P_1 \oplus P_2) \neq S\,(P_1) \oplus S(P_2)$$

where P_1 *and* P_2 are plaintexts.

For example,

Let the permutation, $\pi = \{2, 1, 4, 3\}$,

Plaintext $P_1 = \{1, 1, 0, 0\}$,
Plaintext $P_2 = \{0, 0, 1, 1\}$,
Then $S\,(P_1) = \{0, 0, 1, 1\}$,
$S\,(P_2) = \{1, 1, 0, 0\}$,
$P_1 \oplus P_2 = \{1, 1, 1, 1\}$,
$S\,(P_1 \oplus P_2) = \{0, 0, 0, 0\}$,
and $S\,(P_1) \oplus S\,(P_2) = \{1, 1, 1, 1\}$

Therefore, $S\,(P_1 \oplus P_2) \neq S\,(P_1) \oplus S(P_2)$

Therefore S is non-linear.

6. 3 Theorem 2: (Resistance against differential crptanalysis)

For the 2^n plaintext pairs of inputs with the input difference ΔP, the corresponding cipher pairs satisfy the relation

$\Delta P = \Delta C$. So, there exists only one plaintext pair (P_1, P_2) with $1/2^n$ probability (almost zero for large n) such that the two plaintexts are related by $P_1 \oplus \Delta P = P_2$. For an n-bit input function, our algorithm has $1/2^n$ probability to allow differential attack. So differential attack requires more work to determine the key which is equivalent to brute forcing the key.

6. 4 Statistical Analysis

A statistical analysis was carried out by evaluating the output for *8-bit blocks*. Future work will address the larger block sizes. A complete data set is obtained with all the possible input parameters and processed and analyzed to ensure accuracy. A complete data set is selected because it is useful in evaluating the randomness of the proposed algorithm. To examine the sensitivity of algorithm, the following input parameters were considered:

Permutation of integer sequence: A complete 8! (40,320 data set) integer sequence is constructed.

Round keys and initialization vector: For each permutation, there were 3 round keys generated and choose any one of the key as the initialization vector. So, for 40,320 sequences, 1, 20, 960 keys were constructed.

Plaintext block: A complete 256 sequences (2^8) were examined.

Plaintext/Ciphertext correlation:

In order to examine the randomness of the ciphertext (based on the plaintext and random permutation), 256 sequences were constructed. Each sequence was a result of the output of the algorithm of 256 *ciphertext blocks* using 256 plaintexts for all permutations in the Cipher block chaining (CBC) mode. In order to study the correlation of plaintext/ciphertext pairs, a random permutation (choosing any one sequence) with its corresponding 256 ciphertext blocks is analyzed with all the possible permutations and its corresponding ciphertext blocks were examined for the algorithm. Table 2 shows the number of permutation and ciphertext correlation sequences in the range <0 – 0.9. The numbers of sequences vary greatly between the permutation correlation and the ciphertext correlation and the attacker finds it difficult to predict the correlations. Thus the permutation correlation (key) and the ciphertext correlation in our experiment as analyzed in table 2 show the non-existence of definite correlation between the key and the system output. Therefore, our system is secure against correlation attack.

Table 2: permutation-ciphertext correlation sequences

Parameters range	Number of Permutation correlation sequences	Number of Ciphertext correlation sequences
>0.9	92	7680
>0.8	438	7679
>0.7	1159	7678
>0.6	2315	7677
>0.5	3960	7676
>0.4	6587	7675
>0.3	9304	7674
>0.2	12474	7765
>0.1	15979	9847
<0	19692	16304

7.0 CONCLUSION

In this paper, a novel full encryption approach designed to accommodate fast performance and security for real-time secure video communication applications is proposed. Based on the effective substitution and diffusion components, the system encrypts and decrypts data stream using a pseudorandom integer sequence as the key. Experimental results showed that the proposed algorithm works faster than AES and RC6 in terms of encryption time and frame rate and thus suitable for rendering the data in real-time by the receiver's playback. Histogram analysis, correlation analysis and resistance against differential cryptanalysis proved the efficiency of the cipher. It can even be used to improve the data security of algorithms as previously proposed [34, 35, 36].

REFERENCES

[1] William Stallings, *Cryptography and Network security : Principles and practice, 3/E.* Pearson Education, 2003.

[2] S. Wail Elkilani, M. Hatem M. Abdul-Kader, 2009. Performance of Encryption Techniques for Real Time Video Streaming, *International Conference on networking and Media Convergence, pp. 130-134.*

[3] S. A Aly and E. Al-Shaer, 2003. A Light-Weight Encrypting For Real Time Video Transmission, *CTI Symposium Conference, DePaul University,* Chicago, USA.

[4] C. E. Shannon,1949. Communication Theory of Secrecy System, Bell System Technical Journal, Vol.28, No. 4, pp. 656-715.

[5] Chung-ming ou, *2008*. Design of block ciphers by simple chaotic functions. *IEEE computational Intelligence magazine, ,* Vol.3, No. 2, pp. 54-59.

[6] Xun Yi, Chik How Tan, Chee Kheong Siew, 2002. A New Block Cipher Based on Chaotic Tent Maps. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications,* Vol. 49, No. 12, pp. 1826 – 1829.

[7] N. K. Pareek, Patidar Vinod, K. K. Sud, 2005. Cryptography using multiple one-dimensional chaotic maps. *Communications in Nonlinear Science and Numerical Simulation,* Vol. 10, pp. 715-723.

[8] Wang Xing-Yuan, Yu Qing, 2009. A block encryption algorithm based on dynamic sequences of multiple chaotic Systems, *Communications in Nonlinear Science and Numerical Simulation,* Vol. 14, pp. 574-581.

[9] Chang-Doo Lee, Bong-Jun Choi, Kyoo-Seok Park, 2004. Design and evaluation of a block encryption algorithm using dynamic key mechanism. *Future Generation Computer Systems,* Vol. 20, pp. 327-338.

[10] N. K. Pareek, Patidar Vinod, K. K. Sud, 2003. Discrete chaotic cryptography using external key. *Phys. Lett.* A, Vol. 309, pp. 75-82.

[11] National Institute of Standards and Technology, 2001. *Advanced Encryption Standard*, FIPS 197. en.wikipedia.org/wiki/Triple_DES.

[12] J. Adam Slagell, 2004. Known-plain text attack against a permutation based video encryption algorithm, Available from http://eprint.iacr.org/2004/011.pdf.

[13] L. Tang, 1996. Methods for encrypting and decrypting MPEG video data efficiently, in *Proceedings of the Fourth ACM International Multimedia Conference* (ACM Multimedia'96), Bosten, MA, pp. 219.

[14] L. Qiao, K. Nahrstedt, 1997. A new algorithm for MPEG video encryption, in *Proceedings of the First International Conference on Imaging Science, Systems and Technology,* CISST' 97, Las Vegas, Nevada, pp. 21-29.

[15] C. Shi, B. Bhargava, 1998. A Fast MPEG video encryption algorithm, in *Proceedings of the 6th International Multimedia conference,* Bristol, UK, pp.12-16.

[16] C. Shi, S. Wang, B. Bhargava, 1999. MPEG video encryption in real-time using secret key cryptography, in *proceedings of the International conference on parallel and distributed processing techniques and applications,* pp.191-201.

[17] T. B. Maples, G. A. Spanos, 1995. Performance study of selective encryption scheme for the security of networked real-time video, in *proceedings of the 4th International conference on computer and communications,* Las Vegas, NV, pp. 2-10.

[18] G. A. Spanos, T. B. Maples, 1996. Security for Real-time MPEG compressed video in distributed multimedia applications, *proceedings of the Conference on computers and communications,* pp. 72-78.

[19] T. Lookabaugh, D. C. Sicker, 2004. Selective encryption for consumer applications. *IEEE communications magazine,* Vol.42, No. 5, pp. 124-129.

[20] Franco Chiaraluce, Lorenzo CiccarellI, Ennio gambi, Paola Pierleoni, Maurizio Reginelli, 2002. A new chaotic algorithm for video encryption, *IEEE Transactions on Consumer Electronics*, Vol.48, No. 4, pp. 838-844.

[21] Fang Shang, Kehui Sun, Yongqi Cai, 2008. An efficient MPEG video encryption scheme based on chaotic cipher, *IEEE Congress on Image and Signal Processing*, pp. 12-16.

[22] Hephzibah Kezia, Gnanou Florence Sudha, 2008. Encryption of digital video based on Lorenz Chaotic System, *IEEE Advanced Computing and Communications*, pp. 40-45.

[23] Shuhui Chen, Zengqiang Chen, Zhuzhi Yuan, 2008. A compound video encryption algorithm based on Hyperchaos, *IEEE conference on Innovative Computing information and Control*, pp. 560-563.

[24] Shujun Li, Xuan Zheng, Xuangin Mou And Yuanlong Cai, 2002. Chaotic encryption scheme for Real-timecdigital video, *proceedings of SPIE*, Vol. 4666, pp. 149-160.

[25] Ajay Kulkarni, Saurabh Kulkarni, Ketki Haridas, Aniket More, 2013. Proposed Video Encryption Algorithm v/s Other Existing Algorithms: A Comparative Study, *International Journal of Computer Applications,* Vol. 65, No. 1, pp.1-5.

[26] Euijin Choo, Jehyun Lee, Heejo Lee, Giwon Nam, 2007. SRMT: A Lightweight Encryption Scheme for Secure Real-time Multimedia Transmission, *IEEE International Conference on Multimedia and Ubiquitous Engineering, pp. 60-65.*

[27] C. Narsimha Raju, Ganugula Umadevi, Kannan Srinathan and C. V. Jawahar, 2008. Fast and Secure Real-Time Video Encryption, in *Sixth Indian Conference on Computer Vision, Graphics & Image Processing*.

[28] Ahmad H. Omari, Basil M. Al-Kasasbeh, Rafa E. Al-Qutaish and Mohammad I. Muhairat, 2009. DEA-RTA:

[29] A Dynamic Encryption Algorithm for the Real-Time Applications, *international journal of computers*, Vol. 3, pp. 191- 1999.

[30] Abdulkarim Amer Shtewi†, Bahaa Eldin M. Hasan, Abd El Fatah .A. Hegazy, 2010. An Efficient Modified advanced Encryption Standard (MAES) Adapted for Image Cryptosystems, *International Journal of Computer Science and Network Security*, Vol.10 No.2.

[31] Xun Yi, Chik How Tan, Chee Kheong Siew And Mahbubur Rahman Syed, *2001.* Fast encryption for multimedia, *IEEE Transactions on Consumer Electronics,* Vol. 47, No. 1, 101-107.

[32] N. Radha and M. Venkatesulu, 2012. A Chaotic Block Cipher for Real-Time Multimedia, *Journal of Computer Science*, Vol. 8, No. 6, pp. 994-1000.

[33] http://en.wikipedia.org/wiki/Confusion_and_diffusion.

[34] Raj, R.G. and Abdul-Kareem, S. (2009), Information Dissemination And Storage For Tele-Text Based Conversational Systems' Learning. Malaysian Journal of Computer Science, Vol. 22(2): Dec 2009. pp 138-159.

[35] Raj, R.G. and Abdul-Kareem, S. (2011). A Pattern Based Approach for The Derivation Of Base Forms Of Verbs From Participles And Tenses For Flexible NLP. Malaysian Journal of Computer Science, Vol. 24(2): Jun 2011. pp 63-72.

[36] Raj, R.G. and Balakrishnan, V. (2011). A Model For Determining The Degree Of Contradictions In Information. Malaysian Journal of Computer Science, Vol. 24(3): September 2011. pp 160-167